

Risk Assessment and Collaborative Information Awareness for Plan Execution

Andrei Tudor Soeanu Caval

A Thesis
In
The Concordia Institute
for
Information Systems Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of
Doctor of Philosophy (Information and Systems Engineering) at
Concordia University
Montreal, Quebec, Canada

March 2019

© Andrei Tudor Soeanu Caval, 2019

CONCORDIA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: **Andrei Tudor Soeanu Caval**

Entitled: **Risk Assessment and Collaborative Information Awareness for Plan Execution**

and submitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy (Information and Systems Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____	Chair
Dr. Mojtaba Kahrizi	
_____	External Examiner
Dr. Khalil El-Khatib	
_____	External to Program
Dr. Otmane Ait Mohamed	
_____	Examiner
Dr. Anjali Awasthi	
_____	Examiner
Dr. Chun Wang	
_____	Thesis Supervisor
Dr. Mourad Debbabi	

Approved by _____
Dr. Chadi Assi, Graduate Program Director

Thursday, May 2, 2019 _____
Dr. Amir Asif, Dean
Gina Cody School of Engineering and Computer Science

ABSTRACT

Risk Assessment and Collaborative Information Awareness for Plan Execution

Andrei Tudor Soeanu Caval, Ph.D.

Concordia University, 2019

Joint organizational planning and plan execution in risk-prone environment, has seen renewed research interest given its potential for agility and cost reduction. The participants are often asked to quickly plan and execute tasks in partially known or hostile environments. This requires advanced decision support systems for situational response whereby state-of-the-art technologies can be used to handle issues such as plan risk assessment, appropriate information exchange, asset localization and adaptive planning with risk mitigation. Toward this end, this thesis contributes innovative approaches to address these issues, focusing on logistic support over risk-prone transport network as many organizational plans have key logistic components. Plan risk assessment involves property evaluation for vehicle risk exposure, cost bounds and contingency options assessment. Appropriate information exchange involves participant specific shared information awareness under unreliable communication. Asset localization mandates efficient sensor network management. Adaptive planning with risk mitigation entails limited risk exposure replanning, factoring potential vehicle and cargo loss. In this pursuit, this thesis first investigates risk assessment for asset movement and contingency evaluation using probabilistic model-checking and decision trees, followed by elaborating a gossip based protocol for hierarchy-aware shared information awareness, also assessed via probabilistic model-checking. Then, the thesis proposes an evolutionary learning heuristic for efficiently managing sensor networks constrained in terms of sensor range, capacity and energy use. Finally, the thesis presents a learning based heuristic for cost effective adaptive logistic planning with risk mitigation. Instructive case studies are also provided for each contribution along with benchmark results evaluating the performance of the proposed heuristic techniques.

ACKNOWLEDGEMENTS

I would like to hereby acknowledge the valuable support that I have received in my pursuit of research activities, which culminated with the present thesis. First, I would like to express my utmost gratitude to my supervisor Dr. Mourad Debbabi. Specifically, his vast knowledge and extensive teaching experience along with his guidance, patience and support allowed for the completion of my thesis research work. In addition, I would like to thank all the committee members of my dissertation, namely Dr. Khalil El-Khatib, Dr. Otmane Ait Mohamed, Dr. Anjali Awasthi and Dr. Chun Wang for evaluating my thesis.

I would also like to extend my thanks to my family, friends and colleagues for all their help during challenging times as well as for their constant encouragement and kindness. A special thanks goes to Dr. Sujoy Ray for his collaboration and valuable feedback. Moreover, I would like to extend my thanks to several esteemed researchers from Defence Research & Development Canada, including Dr. Mohamad Allouche, Dr. Abdeslem Boukhtouta, Defence Scientist Jean Berger, Dr. Micheline Belanger and Dr. Nicolas Lechevin, all of whom I had the pleasure to collaborate with over the past years.

Furthermore, I would like to express my gratitude to the Natural Sciences and Engineering Research Council of Canada (NSERC), the Department of National Defence (DND), MDA Corporation and Concordia University for their financial support to conduct my thesis research. In addition, I sincerely thank Concordia Institute for Information Systems Engineering (CIISE) for the support provided, which notably contributed to successfully carrying out this research work.

TABLE OF CONTENTS

List of Figures	viii
List of Tables	xi
Chapter 1. Introduction	1
1.1 Motivations	2
1.2 Transportation Risk Analysis	3
1.3 Collaborative Plan Execution Monitoring	4
1.4 Sensor Network Management for Asset Localization	5
1.5 Logistic Planning with Risk Mitigation and Plan Adaptation	6
1.6 Objectives	7
1.7 Contributions	8
1.8 Document Organization	9
Chapter 2. Background and Related Work	10
2.1 Monitoring Techniques Relevant for Plan Execution	10
2.2 Transport and Plan-related Risk Assessment	13
2.2.1 Model-based Formal Analysis and Decision Tree-based Approaches . . .	13
2.2.2 Simulation and Network-Flow Methods	15
2.2.3 Fuzzy Logic, Bayesian Reasoning and Analytic Assessment Techniques .	16
2.3 Information Sharing for Plan Execution Monitoring	17
2.3.1 Data Aggregation and Exchange Protocols	20
2.3.2 Model-based Formal Protocol Assessment	20
2.3.3 Network-oriented and Analytic Assessment Techniques	21
2.4 PRISM Model-Checker and Markov Decision Processes	22
2.5 Asset Localization and Sensor Networks	25
2.5.1 Localization Techniques for Supply Chain Management and Monitoring .	27
2.5.2 Localization using Wireless Sensor Networks	28
2.5.3 Sensor Network Focus of Attention and Target Assignment	29
2.6 Logistic Support Planning and Risk Mitigation	30

2.6.1	Facility Location and Vehicle Routing	32
2.6.2	Risk Mitigation	33
2.7	Summary	35
Chapter 3. Risk Analysis Using Probabilistic Model-Checking		36
3.1	Decision Trees	37
3.2	System Description	38
3.3	Assumptions	38
3.4	Approach	39
3.5	Static and Dynamic Entities	40
3.6	Case Study on Decision Making based on Decision Trees	44
3.7	Property Specification	51
3.8	Summary	55
Chapter 4. Hierarchy-Aware Distributed Plan Execution Monitoring		57
4.1	Assumptions	59
4.2	Approach	59
4.3	Asymmetric Clustering for Hierarchical Command Chain	61
4.4	Protocol Formalization	62
4.5	Information Awareness Degree	65
4.6	Case Study on Assessing Shared Information Awareness	70
4.7	Summary	77
Chapter 5. Efficient Sensor Network Management for Asset Localization		79
5.1	Context and Assumptions	80
5.2	Problem Description and Modelling	81
5.2.1	Sensor Focus of Attention	81
5.2.2	Problem Statement	81
5.3	Proposed Approach	85
5.4	Case Study	89
5.5	Benchmark Results	102

5.5.1	Parameter Exploration and Performance Assessment	103
5.5.2	Result Analysis	106
5.5.3	Impact of Energy Budget Restriction	108
5.6	Summary	112
Chapter 6. Logistic Planning with Risk Mitigation and Plan Adaptation		113
6.1	Risk Constrained Multi-Depot Vehicle Routing Problem	114
6.1.1	Overview of the Problem	114
6.1.2	Assumptions	115
6.1.3	Risk on Route Evaluation Technique	116
6.2	Problem Modelling	118
6.2.1	Variables and Parameters	121
6.2.2	Non-Linearity Aspect	122
6.2.3	Linear Programming Model using Additive Risk Approximation	123
6.3	Solution Approach	127
6.3.1	Algorithm	128
6.3.2	Discussion	131
6.4	Case Study	134
6.4.1	Route Generation on Low Risk of Vehicle Failure	135
6.4.2	Route Generation on High Risk of Vehicle Failure	141
6.4.3	Route Replanning	148
6.5	Benchmark Results	150
6.6	Summary	154
Chapter 7. Conclusion		156
Bibliography		160
Appendix A		177

List of Figures

2.1	Categorization of noteworthy initiatives in the scope of “Transport and Plan-related Risk Assessment”	14
2.2	Categorization of noteworthy initiatives in the scope of “Information Sharing for Plan Execution Monitoring”	19
2.3	Categorization of noteworthy initiatives in the scope of “Asset Localization and Sensor Networks”	26
2.4	Categorization of noteworthy initiatives in the scope of “Logistic Support Planning and Risk Mitigation”	31
3.1	Decision tree example	37
3.2	Case study	38
3.3	Risk assessment technique	45
3.4	Decision tree with EMC	47
3.5	Decision tree with cost threshold	48
3.6	PRISM model-checker environment - Module editor	52
3.7	PRISM model-checker environment - Property specification and verification . . .	52
3.8	PRISM model-checker environment - Property verification results	53
3.9	PRISM model-checker environment - Verification of modified properties	54
4.1	Example of hierarchical clustering for collaborative plan execution	58
4.2	Participant threads/interactions (a) and event dissemination fresh window (b) . .	60
4.3	Symmetric membership across clusters (a) and symmetric/asymmetric membership in the same cluster (b)	61
4.4	State diagrams corresponding to the participants in cluster (A^s, B^s, C^a)	67
4.5	State diagram corresponding to the system fresh window size update	68
4.6	Probability assessment example	69
4.7	Asymmetric clustering for hierarchical information dissemination	70

4.8	Probability assessment on achieving fully shared information awareness for participants C and D	71
4.9	Probability assessment on achieving partially shared information awareness for participants C and D	72
4.10	Probability to achieve full/partial information awareness for $fws = 5$ and increasing error rate levels	73
4.11	Participants changing position and forming new cluster arrangements	74
4.12	Comparison of shared information awareness in different cluster arrangements . .	75
4.13	Comparison of cubic and higher degree polynomial approximations of shared information awareness for two cluster arrangements	76
5.1	Target T inside the triangle delimited by $\{S_A, S_B, S_C\}$ and marked intersection area for possible error in target identification	85
5.2	Case study problem (left), nearest neighbour solution cost=2742.1184 (middle), optimal solution cost=392.1358 (right)	90
5.3	Breakdown cost comparison between the nearest neighbour solution (a) and the optimal solution (b)	91
5.4	Heuristic solution convergence profile for increasing number of sampled combinations per iteration	94
5.5	Solution cost comparison for increasing energy budget (8, 9, 10, 11)	97
5.6	Solution characteristics for increasing energy budget relative to ‘per target focusing sensor count’ and ‘per target solution cost’	98
5.7	Solution characteristics for increasing energy budget relative to ‘sensor use count’ and ‘network capacity use ratio’	99
5.8	Solution comparison (sample cnt. 40/50) for increasing energy budget relative to ‘network capacity use ratio’ and ‘solution cost’	101
5.9	Performance assessment for the 20, 23, 40 and 45 node problems where the maximum sensor capacity is 4	104
5.10	Performance assessment for the 50, 51, 55 and 60 node problems where the maximum sensor capacity is 5	105

5.11	Performance assessment for the 65, 70, 76 and 101 node problems where the maximum sensor capacity is 6	105
5.12	Average sensor capacity use relative to the benchmark results obtained for each problem instance with different energy budgets	110
6.1	Risk on route example	116
6.2	Running example with 3 depots and 13 demand nodes (a) and solution illustration with virtual nodes D0 and 17 as depot node and sink node for vehicle routes (b)	119
6.3	Additive risk approximation (solid line) vs. actual cumulative risk (dashed line)	123
6.4	Solution generation procedure	127
6.5	Risk on route solution sensitivity (1% leg risk, without max. vehicle risk, solutions still under threshold; 2% leg risk, without and with max. vehicle risk)	136
6.6	Low risk on route solution details without and with additive risk approximation	138
6.7	Solution cost analysis for low risk of traversal	139
6.8	Solution sensitivity cost breakdown without and with (marked by [...]) risk approximation: 1% leg risk, solutions still under threshold and 2% leg risk, max. vehicle risk applied	140
6.9	Risk on route solution sensitivity (10% leg risk, max. vehicle risk not applied)	142
6.10	High risk on route solution details	144
6.11	Risk on route solution sensitivity (10% leg risk, with max. vehicle risk; 20% leg risk, without and with max. vehicle risk)	145
6.12	Solution cost analysis for high risk of traversal	146
6.13	Solution cost breakdown and comparison for risk on route sensitivity	147
6.14	Solution cost breakdown and comparison for selected risk on route solutions	147
6.15	Proactive and reactive risk mitigation replanning	149

List of Tables

2.1	Noteworthy approaches or techniques in the interest area of “Transport and Plan-related Risk Assessment”	15
2.2	Noteworthy approaches or techniques in the interest area of “Information Sharing for Plan Execution Monitoring”	18
2.3	Noteworthy approaches or techniques in the interest area of “Asset Localization and Sensor Networks”	27
2.4	Noteworthy approaches or techniques in the interest area of “Logistic Support Planning and Risk Mitigation”	32
3.1	Routing-related risk probabilities	39
3.2	Probability utility	50
4.1	Identification of best fresh window size relative to acceptable probability values over a threshold of 50%	76
5.1	Asset localization case study problem details	90
5.2	Sensor-target assignment combinations and breakdown of the received votes	92
5.3	Elite solutions for the first iteration, with most amount of votes (5) provided by 5 solutions for combination [Q, S] of sensor d	93
5.4	Pruned combinations over successive iterations	95
5.5	Benchmark problems	102
5.6	Benchmark results for asset localization	107
5.7	P-n20 (100% energy budget)	110
5.8	P-n20 (75% energy budget)	110
5.9	P-n20 (50% energy budget)	111
6.1	Benchmark results for RCMDVRP	151
6.2	Vehicle risk exposure for RCMDVRP solutions	152

6.3 Best MDVRP solution evaluation in presence of risk vs. the savings achieved by the best RCMDVRP solutions	153
A.1 Data and target assignment solution for problem P-n23	177
A.2 Data and target assignment solution for problem P-n23 continued	177
A.3 Data and target assignment solution for problem P-n40	178
A.4 Data and target assignment solution for problem P-n40 continued	178
A.5 Data and target assignment solution for problem P-n45	179
A.6 Data and target assignment solution for problem P-n45 continued-1	179
A.7 Data and target assignment solution for problem P-n45 continued-2	180
A.8 Data and target assignment solution for problem P-n50	180
A.9 Data and target assignment solution for problem P-n50 continued-1	181
A.10 Data and target assignment solution for problem P-n50 continued-2	181
A.11 Data and target assignment solution for problem P-n51	182
A.12 Data and target assignment solution for problem P-n51 continued-1	182
A.13 Data and target assignment solution for problem P-n51 continued-2	183
A.14 Data and target assignment solution for problem P-n55	183
A.15 Data and target assignment solution for problem P-n55 continued-1	184
A.16 Data and target assignment solution for problem P-n55 continued-2	184
A.17 Data and target assignment solution for problem P-n60	185
A.18 Data and target assignment solution for problem P-n60 continued-1	185
A.19 Data and target assignment solution for problem P-n60 continued-2	186
A.20 Data and target assignment solution for problem P-n65	186
A.21 Data and target assignment solution for problem P-n65 continued-1	187
A.22 Data and target assignment solution for problem P-n65 continued-2	187
A.23 Data and target assignment solution for problem P-n65 continued-3	188
A.24 Data and target assignment solution for problem P-n70	188
A.25 Data and target assignment solution for problem P-n70 continued-1	189
A.26 Data and target assignment solution for problem P-n70 continued-2	190
A.27 Data and target assignment solution for problem P-n70 continued-3	191

A.28 Data and target assignment solution for problem P-n76	192
A.29 Data and target assignment solution for problem P-n76 continued-1	193
A.30 Data and target assignment solution for problem P-n76 continued-2	194
A.31 Data and target assignment solution for problem P-n76 continued-3	195
A.32 Data and target assignment solution for problem P-n101	196
A.33 Data and target assignment solution for problem P-n101 continued-1	197
A.34 Data and target assignment solution for problem P-n101 continued-2	198
A.35 Data and target assignment solution for problem P-n101 continued-3	199
A.36 Data and target assignment solution for problem P-n101 continued-4	200

Chapter 1

Introduction

The expansion of worldwide connectivity brought about significant structural changes in terms of size and complexity of the private and public organizations. Ongoing developments in the global arena require many organizations to carry out plans and operations that need to consider uncertain and potentially hostile environments. Moreover, many organizations collaborate in the execution of joint plans and operations taking place remotely in presence of exogenous events.

Many organizational plans include logistic support components that represent essential aspects in a wide area of activities for which the expected outcomes depend on the success or failure to carry out the underlying transport tasks. Thus, transportation activities can significantly benefit from performing risk analysis on the transport network. Since different routing choices are expected to result in different risk exposure levels, it is necessary to evaluate and if possible mitigate the risk levels associated with different routes as part of planning and decision making activities. Large organizations typically address their logistics sustainment at three different hierarchical levels. These levels are often referred as: Strategic, Operational and Tactical [107]. At the highest level, decision makers evaluate strategies for achieving their goals such as selecting which depots to use for logistic support. Decision makers at operational level analyse the underlying problem and generate solutions (e.g., vehicle routes for commodity delivery) to be implemented at tactical level in order to attain the goals as required. From an organizational command and control perspective, collaborative execution of plans in dynamic, uncertain and potentially hostile environments mandates capabilities to jointly monitor and adapt the plan during execution. In this

setting, the monitoring process resides at operational level while the adaptation takes place at tactical level following instructions received from the operational level, based on strategic policies and directions. This requires appropriate information sharing over noisy communication channels across the chain of command. Moreover, another important aspect is to obtain location information on the plan executing assets as well as on potential plan interfering assets.

In the context of logistic support in a hostile environment, where deployed vehicle assets are exposed to risk on route, risk constrained planning and plan adaptation involves the generation and update of cost effective vehicle routes with risk mitigation. This requires taking into account the potential cost of lost vehicles and failed cargo deliveries.

In contrast to traditional monitoring, which seeks a trade-off between timely response and thorough data gathering and analysis, distributed monitoring involves distributed nodes that exchange gathered data and combine their analysis results in order to present a joint response. This can be particularly suitable when information is shared in an environment prone to communication disruption and exogenous events. While a centralized coordination center can be used in such context, it exhibits a characteristic single point of failure and a potential information bottleneck.

1.1 Motivations

Recent trends toward collaborative involvement of multiple organizations in the execution of operational plans generated significant research interest in assessing and addressing the risk associated to the occurrences of exogenous events.

Past incidents provide a glimpse on the impact resulting from exogenous events such as adverse or extreme weather that can increase the risk for plan execution especially when there is a logistic component involved, which many plans typically have. Notable examples are represented by hurricanes Katrina¹ in 2005 and Maria² in 2017, both of which severely affected the transport infrastructure after landfall, thereby disrupting the relief efforts. Another significant incident³ happened in 2010, when the ash ejected from the eruption of a volcano in Iceland forced many European countries to close their airspace, resulting in a massive disruption of air traffic.

¹<http://bit.ly/2WfCLU3>, Last accessed on 1 March 2019.

²<http://bit.ly/2JA14f9>, Last accessed on 1 March 2019.

³<http://bit.ly/2U1eq78>, Last accessed on 1 March 2019.

Operational plans involving commodity delivery, have goals, allocated resources and tasks that need to be completed in order to achieve the goals. In a disruptive environment, during the plan execution, it may be required to change the course of actions by adapting to the exogenous events that may be encountered. Heuristic techniques are usually employed in order to solve commodity delivery problems typically with an objective of minimizing the routing cost as well as risk exposure when the involved routes have associated risk factors. After the planning processes, a set of tasks is generated where each task can represent for instance a particular vehicle route. Tactical officers are then responsible to execute these tasks. For instance, in case of aid provisioning after a natural disaster or as a result of reconstruction and development initiatives, programs such as United Nations Development Assistance Plan (UNDAP) or United Nations Development Assistance Framework⁴ (UNDAF) may be employed to launch major operations. For example, in the aftermath of an earthquake or a hurricane, disaster management teams decide over depot establishment, while the aid delivery and rescue teams decide on vehicle routing taking into account the costs and risks involved for tasking specific vehicles. Then, designated rescue officers execute the corresponding tasks using the vehicles available at the depots.

An appropriate hierarchy-aware information sharing technique can provide effective support for addressing plan execution events (e.g., vehicle failure) in the environment of the participants. This entails achieving high level of common knowledge on the same levels in hierarchy (e.g., rescue officers) and aggregated information across the hierarchical levels (e.g., disaster management teams). As such, monitoring plan execution and the events in the environment is essential for understanding the current state of the ongoing plan tasks. This requires protocols, algorithms and analysis techniques that can support collaborative planning and plan execution monitoring.

1.2 Transportation Risk Analysis

From a conceptual standpoint, risk is traditionally conceived as “reflecting variation in the distribution of possible outcomes, their likelihoods, and their subjective values” [95]. Moreover, according to Beatriz *et al.* [54], in relation to the OHSAS 18001 standard, risk is defined as: “the

⁴<http://bit.ly/2FsZzKE>, Last accessed on 1 March 2019.

combination of the likelihood of an occurrence of a hazardous event or exposure(s) and the severity of injury or ill health that can be caused by the event or exposure(s)". In addition, according to "Transport Canada", the underlying concepts relate to the likelihood that something unwanted is going to happen, the resulting consequences if it does, and the outcome uncertainty. In this setting, risk management brings the idea that the likelihood of an unwanted event happening can be reduced, or its consequences minimized. In essence, risk management for logistic plans aims at reducing accident likelihood and severity of failed deliveries while keeping cost under acceptable threshold. Notable benefits can result from such risk management. It supports strategic and organizational planning and allows decision makers to handle uncertainty (unexpected or unwanted events). It also allows getting hold of opportunities resulting from risk mitigation, and enhances stakeholders' communication among other things as outlined by Richardson *et al.* in [123].

Since different route choices can result in different risk exposure levels it is necessary to evaluate and address the risk exposure levels associated with different routes as part of planning and decision making activities. In this pursuit, this thesis investigates a risk assessment technique based on formal analysis of transportation systems using probabilistic model-checking.

1.3 Collaborative Plan Execution Monitoring

Collaborative planning and plan execution monitoring represent noteworthy trends both in national and international context, potentially involving different governmental organizations. This promotes a synergy resulting from the specific area of specialization of various organizations working together toward a common goal.

In international arena, multi-national coalitions can benefit from sharing resources and information in order to reduce operational cost while achieving increased agility during disaster relief operations and humanitarian missions. Also, at national level, various agencies can inter-operate in order to carry out development and contingency plans in situations of crisis. In this respect, effective information sharing plays an important role for replanning (adaptive planning) in relation to a broad range of operational plans that are executed in uncertain and potentially hostile environments. This allows to provide appropriate response when faced with evolving circumstances. The potential changes in transport infrastructure due to natural disasters, emergency

situations, etc., along with potential disruptions of communications, represent significant challenges for executing plans that depend on logistic support components. This highlights the importance of having adequate supplies for conducting operations along with a corresponding level of information awareness. Thus, effective decision support mechanisms encompassing such concerns are needed for operational and logistic planning and execution. Given that monitoring mandates an effective reporting component, in the context of a command hierarchy, an important aspect is to assure that reporting occurs accordingly. Here we distinguish between command information flow that propagates from higher to lower levels and reporting, which represents a flow of information from the bottom of the hierarchy toward the top.

In this context, this thesis investigates a hierarchy-aware distributed monitoring procedure applicable on generic operational plans that require up-to-date shared information awareness.

1.4 Sensor Network Management for Asset Localization

Plan execution often involves the use of various assets, including transport vehicles, which are needed to carry out plan related tasks such as the provision of logistic support. When such tasks take place over potentially perilous terrains where hostile actors may be present, tasked assets are often required to keep a low profile by limiting radio transmission thereby reducing risk exposure. Notwithstanding, such assets can still receive plan execution updates and instructions via passive radio reception. In this setting, decision makers and stakeholders can greatly benefit from asset localization capabilities involving wireless sensor networks (WSN), which do not rely on asset self reporting. Another benefit of using sensor networks for asset localization relates to having the possibility to also detect and localize assets that belong to other stakeholders.

Asset localization through WSN represents a two part problem: First, we have the aspect of sensor assignment for each target asset, typically referred to as the sensor focus of attention [72]. Second, we have the aspect of location identification for each target asset, which involves deriving target locations from a given assignment of sensors to targets. The latter problem is approached in our scope of interest by triangulation given that we consider omnidirectional sensors with different sensing ranges and target focusing capacity. Thus, we consider that at least three different sensors need to be assigned to a target asset in order to properly localize it [90].

In the foregoing context, FOA and related sensor assignment problems represent key aspects for effectively employing a sensor network, particularly when heterogeneous sensors can focus on a subset of the targets in their coverage area in each time unit. Moreover, different sensors may have different abilities in terms of detection range (e.g., due to difference in elevation level) and target focusing capabilities (i.e., one or more targets per time unit). In this respect, more capable sensors may be placed in locations deemed of higher importance (e.g., higher trafficability potential). An effective sensor-target assignment allows to minimize the localization error by appropriately assigning sensors to targets. The heterogeneity aspect introduces an extended level of complexity due to the vast number of possible assignment combinations, which requires quick heuristic search methods for near-optimal sensor assignment.

In this setting, this thesis elaborates an effective heuristic procedure for sensor network management directed at finding appropriate multisensor-multitarget assignments, aiming toward minimizing the overall localization error cost.

1.5 Logistic Planning with Risk Mitigation and Plan Adaptation

During the collaborative execution of plans, with associated logistic support components, in potentially hostile environments, assets such as transport vehicles are exposed to various risk factors, which may lead to vehicle loss and cargo delivery failure.

In this setting, the participants can significantly benefit from a decision support system whereby potential vehicle and cargo loss can be mitigated by limiting vehicle risk exposure and by prioritizing larger deliveries over smaller ones while also aiming for reduced routing cost. The underlying solution approach can benefit both at the planning stage as well as part of proactive and reactive plan adaptation activities prompted by the occurrence of exogenous events (e.g., risk factor updates, vehicle losses) during plan execution. When multiple participants are involved, this corresponds to generating cost effective solutions for the related risk constrained multi depot vehicle routing problem. This can also help addressing the specific challenges encountered during the provision of logistic support in perilous environments by quickly providing updated solutions.

In this context, this thesis elaborates a cost effective learning-based heuristic technique to minimize the combined routing cost and potential vehicle loss and cargo delivery failure costs.

1.6 Objectives

This research endeavour envisions the elaboration of specialized approaches underpinning related decision support system framework components suitable for: assessing transport risk for plans with logistic support components; enhancing situational awareness during collaborative monitoring of plan execution; managing sensors networks for improved asset localization during plan execution and logistic plan adaptation with risk mitigation. More specifically, in the scope of the present thesis, the decision support components include the following: plan risk analysis, exemplified from the perspective of logistic support, based on assessing semantic models in the form of Markov Decision Processes (MDPs) that can be subjected to probabilistic model-checking; hierarchy-aware distributed plan execution monitoring via shared information awareness under communication disruption, captured by communicating MDPs that can also be subjected to probabilistic model-checking; resource constrained sensor network management for asset localization using a tailored heuristic technique and adaptive logistic support with risk mitigation also employing a heuristic technique. Thus, the proposed research objectives are as follows:

- Elaborate an approach capturing risk-prone logistic support tasks using MDPs that can be subjected to probabilistic model-checking and assessed thereafter via related decision trees.
- Elaborate a hierarchy-aware information sharing protocol tolerant to transmission error and model it via communicating MDPs, in support of distributed monitoring of plan execution.
- Elaborate an efficient heuristic technique for sensor network management allowing to adequately assign constrained sensors to target-assets, thus minimizing localization error cost.
- Elaborate a heuristic solution technique for risk constrained logistic delivery problems in pursuit of minimizing the combined routing cost and potential vehicle and cargo loss cost.
- Conduct illustrative case studies for each of the elaborated approaches and document benchmark results evaluating the performance of the proposed heuristic techniques.

The envisioned framework components will be very useful for providing decision support on planning, plan analysis, plan execution monitoring as well as plan adaptation.

1.7 Contributions

In the context of collaborative planning and plan execution, the overarching theme of this thesis is addressing the need for logistic support planning, plan execution monitoring and replanning in risk prone and communication disrupted environments. In this setting, the thesis considers four important and complementary capabilities: (1) assessment of risk and contingency options in support of logistic plan analysis; (2) collaborative plan execution monitoring under unreliable communication along with (3) sensor network management for asset localization, both in support of plan execution situational awareness and (4) logistic planning/replanning with risk mitigation based on the obtained situational awareness. Consequently, the present research effort focuses on contributing innovative approaches to provide enhanced decision making support for collaborative planning and plan execution monitoring. The emphasis is on logistic support planning and plan adaptation in potentially hostile environments. As such, this thesis has the following contributions:

- *Risk Analysis and Contingency Options Assessment*: This contribution proposes a procedure that leverages probabilistic model-checking to evaluate the risk and contingency options related to logistic support transportation tasks. In addition, risk related properties can be assessed for probabilistic behavioural models capturing vehicle movement. Alongside, decision trees derived from model-checking results are employed in order to provide insightful means of risk appraisal. Such capabilities allow decision makers to evaluate contingency options and to determine cost bounds for risky logistic support tasks.
- *Hierarchy-Aware Distributed Plan Execution Monitoring*: This contribution involves a distributed and hierarchy-aware monitoring procedure suitable for plan execution taking place in a partially known environment, prone to unreliable communication. The procedure uses asymmetric clustering to reflect hierarchical relationships among participants along with gossip based communication for information exchange. The information sharing mechanism utilizes a fresh information window and the communication among participants is modelled using communicating Markov Decision Processes. This allows assessing the shared information awareness via probabilistic model-checking. In this context, the assessment of formal specifications expressed in probabilistic temporal logic allows to derive the best fresh window size to maximize an information awareness utility function.

- *Efficient Sensor Network Management for Asset Localization:* This contribution introduces a technique allowing to efficiently employ a wireless sensor network (WSN) for asset localization where each target asset needs to be assigned a minimum number (typically three) sensors for proper localization. In this respect, the WSN is composed of heterogeneous sensors with various target focusing capacities and sensing ranges. The problem is further analysed under the constraint of a globally specified overall WSN energy budget, which limits the possible assignments for the capacitated sensors. In this context, a heuristic technique is proposed for efficient sensor to target assignment, leveraging evolutionary learning along with meta-heuristic improvements.
- *Adaptive Logistic Planning with Risk Mitigation:* This contribution presents an approach involving an effective, learning based heuristic solution generation technique for multi-depot logistic planning and plan adaptation in perilous environments where vehicle loss and cargo delivery failure may occur due to risk exposure. The approach allows to generate cost effective and risk limited vehicle routes for the underlying risk constrained multi-depot vehicle routing problem by taking into account the simultaneous mitigation of potential vehicle losses and commodity delivery failure due to on-route risk exposure.

1.8 Document Organization

The remainder of the thesis is organized as follows. In Chapter 2, we discuss relevant initiatives related to the proposed research. Then, Chapter 3 presents the proposed approach for transportation risk analysis via probabilistic model-checking and related decision trees. Subsequently, Chapter 4 elaborates the proposed approach for supporting hierarchy-aware distributed plan execution monitoring and further employs probabilistic model-checking to assess the shared information awareness of plan executing participants. Next, Chapter 5 discusses the proposed heuristic approach on efficient sensor network management for asset localization, which can provide additional support for plan execution monitoring and replanning. Thereafter, Chapter 6 presents the proposed heuristic approach on generating cost effective planning/replanning solutions for risk constrained multi-depot logistic delivery problems. Finally, conclusions are drawn in Chapter 7.

Chapter 2

Background and Related Work

This chapter discusses noteworthy initiatives in the research area of this thesis. As a relevant background, we begin with useful techniques related to plan execution and accompanying logistic support and transport activities in connection to monitoring properties, the information exchanged and corresponding analysis approaches. Next, we continue with a discussion on plan-related transport risk assessment, which can help monitoring activities and plan adaptation by evaluating contingency options. Subsequently, we present several prominent research directions related to plan execution monitoring and information awareness in collaborative setting. Then, given the use of probabilistic model-checking in the contributions of this thesis related to risk assessment and plan execution monitoring, we provide an informative background on the employed probabilistic model-checker, namely PRISM [65]. In addition, we discuss specific details on the selected semantic model and temporal logic among those supported by PRISM. Subsequently, we present a number of relevant works in the field of asset localization. Finally, we discuss various initiatives related to logistic planning and risk mitigation.

2.1 Monitoring Techniques Relevant for Plan Execution

Monitoring techniques generally involve corresponding methodologies for observing the behaviour of the underlying subject processes in order to signal violations of specific properties of interest [148]. In addition, there is a timing perspective that separates different monitoring activities as time-based [68] and event-based [6]. In this regard, time-based monitoring involves suitable

techniques that are leveraging periodic data collection from the running processes, as discussed by Huang *et al.* [68]. In contrast, event-based monitoring mainly refers to the reaction of a monitor to events that are typically generated by changes in the monitored process state space, as detailed by Albari [6]. Moreover, advanced monitoring approaches can also use specification languages in order to describe monitoring properties. As mentioned by Claessen [35], process monitoring can correspond to checking various properties such as safety, liveness, performance, security, etc. In addition, Cormode *et al.* [38] discuss different relevant examples in the context of algorithmic distributed functional monitoring. Moreover, monitoring techniques relevant for aspects such as plan-related policies, exogenous events, fault detection, etc. are discussed by Wilkins *et al.* [148]. Furthermore, the benefits of employing adaptive plan monitoring are discussed by Allen and McCormick [9] with respect to situational awareness and plan adaptation. In this setting, the authors discuss a model-based advisory decision support system. The latter provides agent-based plan monitoring and response capabilities for autonomous decision support and adaptive planning. The advisory system factors potential contingencies along with incoming monitoring data while mitigating limited network bandwidth in remote areas.

In relation to monitoring processes in large networks, the emphasis is on the scalability. A typical approach involves network partitioning into comparable partitions with the least number of interconnections. This corresponds in essence to ensuring that the workload distribution is well balanced. However, Feder *et al.* [52] showed that there is no generally tractable procedure for optimal graph partitioning. This led to the development of heuristic approaches, as discussed by Jarrah and Bard [75], which show the relevance of clustering based partitioning techniques.

Monitoring the execution of operational plans involves the interpretation of the underlying workflow states as well as the assessment of variables related to control functions and global aggregates, as discussed by Fetahi *et al.* in [149]. In centralized setup, one or more collections of monitoring nodes can be used to send information to a central station, as mentioned by Delgado *et al.* in [44]. Agent based collaboration is detailed by Ortiz and Charles in [104] while Kaminka *et al.* [79] discuss agent based monitoring for plan coordination, with many geographically distributed agents that participate in a dynamic environment. The elaboration of suitable algorithms for process execution monitoring has seen a renewed interest in pursuit of scalable, resource-aware data gathering techniques, as described by Shicong *et al.* [129]. The problem of

aggregating the monitored information also involves the ability to handle an increased number of monitoring agents along with the corresponding increase in the communication and processing requirements, as detailed by Rogers *et al.* [124]. In the area of decentralized control of cooperative and autonomous agents, a distributed resource allocation approach is presented by Anussornnitisarn *et al.* [11]. Sharing and combining data also represents an important aspect. In this regard, an effective procedure employs “top- k monitoring queries”, as discussed by Palpanas *et al.* [108]. In essence, the procedure involves the continuous reporting of the k most important parameter values from various distributed data streams.

In a more general context of process execution monitoring, the monitors typically perform a continuous task of ascertaining the condition of an evolving system along with recognizing behavioural anomalies [48, 58, 144]. Nguyen *et al.* [103] present an Online Analytical Processing (OLAP)-oriented architecture for proactive and timely response on unexpected situations, by sensing events in dynamic business environment. OLAP represents a data warehousing technology that can present a large dataset at different abstraction levels, thereby providing valuable insights compared to the bulk data. In relation to logistic support, Hoa *et al.* [67] discuss a mixed OLAP approach that allows to enhance logistics workflow. Also, valuable insights can be obtained by combining data mining with distributed data streams queries, as described by Gang *et al.* [56].

Leveraging the rapid advances in processing, storage and inter-networking, data mining allows to transform large quantities of data into high value information while aiming at scalability. In particular, for logistics planning, data mining allows to obtain fast paths over large routing networks, as detailed by Awasthi *et al.* [15] and data mining was also used by Markovic [96] in the context of predicting stochastic demands. Furthermore, in the context of data mining, as mentioned by Dean and Ghemawat [42], large volumes of data can be reduced to more manageable and meaningful information. This allow decision makers to respond in a more timely and informed manner to evolving situations. In relation to logistics, game theory has also been considered for the interaction of self-interested agents [126] with the objective of obtaining the best strategy that an agent should adopt. This is especially relevant for transportation and supply chain activities potentially involving multiple partners and resource pool fragmentation, as discussed by Xiaolong *et al.* [150]. Moreover, a noteworthy study in relation to incentive alignment for coordination in the context of humanitarian aid distribution has been carried out by Martinez *et al.* [97].

2.2 Transport and Plan-related Risk Assessment

In the context of transportation and related planning activities, the main risk management area of concern involves the reduction of accidents, related penalties and accompanying severity levels, as described by Christopher *et al.* in [33]. Transport planning and management also involve the perception of risk that decision makers might have for different means of transportation, as discussed by Rundmo *et al.* in [125]. An additional aim of the aforementioned work is to examine the relative importance of perception for transport risk factors relative to the perception of non-transport risk factors. In general, when transporting products from different locations (depots) to various destinations (demand points), one has to manage how the products are shipped while assessing, to the extent possible, the potential risks involved, as discussed by Manuj and Mentzer in [94]. In this respect, such assessment can typically be performed as part of business process modelling. Next, we discuss several relevant works in our scope of interest, of which the most noteworthy ones are categorized as shown in Figure 2.1 and Table 2.1.

2.2.1 Model-based Formal Analysis and Decision Tree-based Approaches

Plan specific flow-charts and related activity diagrams can be captured by behavioural models that can be subjected to formal verification as part of business processes modelling. A symbolic model-checking approach for service delivery planning with applicability for search and rescue operations is detailed in [133]. The proposed approach uses the NuSMV¹ model-checker. In the context of systems engineering design, Yosr *et al.* [76] present an automated verification and performance analysis approach for time-constrained activity diagrams described in Systems Modelling Language (SysML) and modelled as discrete time Markov chains. The proposed approach uses the PRISM² probabilistic model-checker.

Decision trees have been used by Berger *et al.* [23] to analyse the risk in a network of suppliers in order to select the most adequate number of suppliers. Decision trees for risk analysis and assessment have been applied by Hulett and Hillson [69, 70] in order to assess the expected value of a project and to identify project alternatives. This contrast to the use of decision trees in the scope of interest of this thesis, which focuses on loss mitigation and potential cost avoidance.

¹<http://nusmv.fbk.eu>, Last accessed on 1 March 2019.

²<http://www.prismmodelchecker.org>, Last accessed on 1 March 2019.

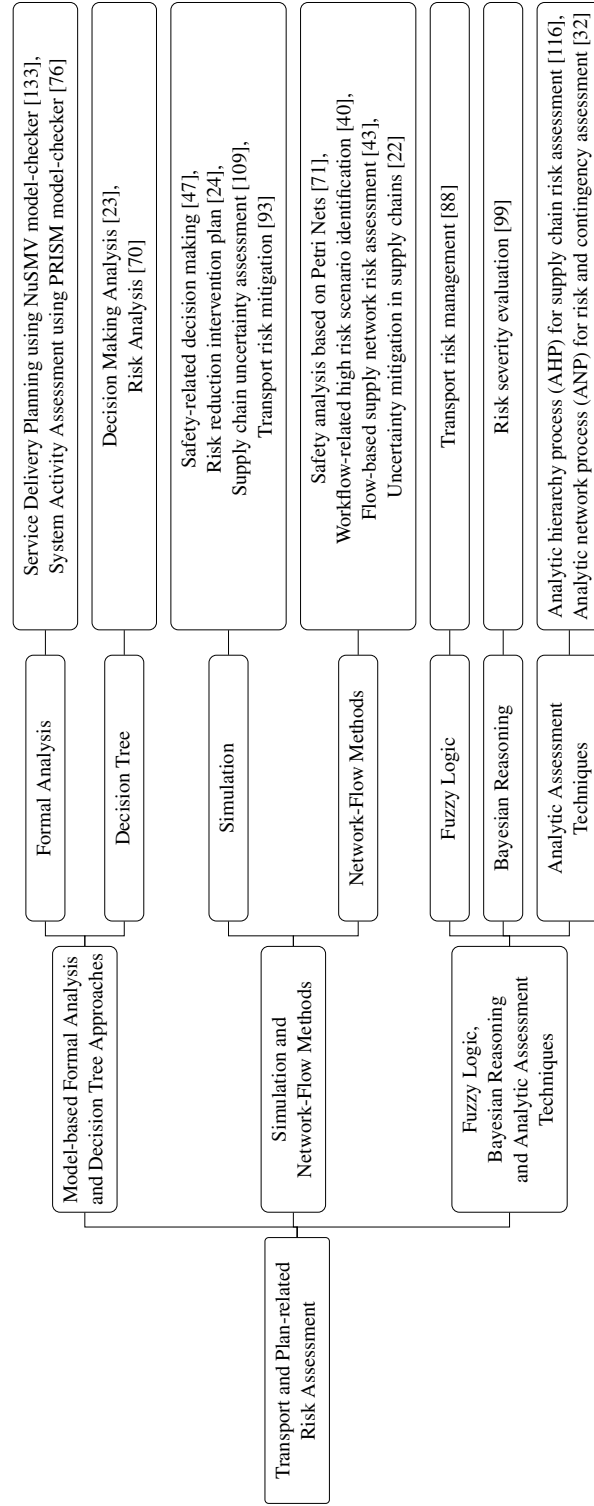


Figure 2.1: Categorization of noteworthy initiatives in the scope of “Transport and Plan-related Risk Assessment”

\ Approach or technique Area of interest \	Formal Analysis	Decision Trees	Simulation Methods	Network-Flow Methods	Fuzzy Logic and/or Bayesian Reasoning	Analytic Assessment
System activity and/or risk scenario and/or risk vs. contingency assessment	[76]			[40]		[32]
Decision analysis and/or risk vs. safety analysis and/or risk severity evaluation		[23, 70]	[47]	[71]	[99]	
Planning for service delivery and/or risk reduction	[133]		[24]			
Supply chain risk and/or uncertainty assessment and/or mitigation			[43, 109]	[22, 43]		[116]
Transportation risk mitigation and/or management			[93]		[88]	

Table 2.1: Noteworthy approaches or techniques in the interest area of “Transport and Plan-related Risk Assessment”

2.2.2 Simulation and Network-Flow Methods

Applied system dynamics can be used to gather insights into potential risk factors that can affect mission critical processes and applications. This technique has been used by NASA to perform risk analysis on the safety-related decision making structure as described by Dulac [47] and also in the manned space program, to improve the understanding of the factors involved in the Columbia shuttle accident. Moreover, stochastic dynamic programming has been applied by Besnard [24] in the context of power systems in order to develop an optimal intervention plan to minimize the risk of equipment failure. Caulkins *et al.* [30] used Integer Programming to find countermeasure combinations with the goal of maximizing system security under fixed resources.

Furthermore, Petri Nets have been used by Iordache and Antsaklis [71] for safety analysis, workflow modelling, verification and authorization and also by Ammar and Leteef [40] in the context of high-risk scenario identification.

A system dynamics model is proposed by Peng *et al.* [109] to analyse the behaviours of a disrupted disaster relief supply chain by simulating uncertainties associated with post-seismic road network and delayed information. This work shows the benefits of strategies considering extrapolation during logistic planning, based on information from both the post-seismic management center and from the affected areas. The paper also proposes the use of decision trees to help decision makers in the planning process. This work employs simulation to ascertain information delay and the impact of changed road conditions on transportation. The use of simulation allows more flexibility in terms of parameter range. However, while computationally more demanding,

model-checking is distinguished by thorough verification of properties expressed in formal logic.

Mahmoudabadi and Abbas [93] address the mitigation of risk factors associated with the transport of hazardous materials. This work proposes the improvement of the underlying transport graph by reducing the risk over the frequently used edges in order to decrease total risk over the transport network. In addition, risk is modelled as a chaotic variable used for the routing problem. A chaotic simulation model is also developed to generate a list of network edges frequently used for transportation while the risk factors over the network are generated using simulation.

The approach for supply network risk assessment presented by Deleris and Feryal [43] rests on a network-flow model combined with Monte-Carlo simulation. This approach includes external events to evaluate uncertainty in supply networks. It accounts for dependencies between products and facilities, enabling a high-level analysis of “loss of product volume” due to network structure and adverse external events. A methodology to generate a robust logistics plan to mitigate demand uncertainty in humanitarian relief supply chains is proposed by Ben-Tal *et al.* [22]. The forenamed work deals with optimizing dynamical assignment of emergency response and evacuation traffic flow. The focus is on mitigating the uncertainty of demand in the aftermath of a disaster.

2.2.3 Fuzzy Logic, Bayesian Reasoning and Analytic Assessment Techniques

Li *et al.* [88] present a tailored fuzzy logic model for risk management in marine oil transport system. The model improves the normal fuzzy expert system via proactive and reactive loops. Experimental results are used to show the benefits in terms of improved risk assessment. This work leverages fuzzy logic, which deals with degrees of truth or relative truths in contrast to probabilistic reasoning, which aims at making predictions about events based on partial knowledge.

The framework discussed by Montewka and Ehlers [99] focuses on ship to ship collisions in open sea. It covers the identification of the events that follow a collision between two ships in open sea, and evaluates the probabilities of these events, concluding by determining the severity of a collision. The developed risk assessment framework is based on Bayesian networks and uses a set of analytical methods for estimating the parameters of the risk model.

The work by Gordana and Vladimir [116] is proposing approaches based on analytic hierarchy process (AHP) and fuzzy AHP used as a tool suite for ranking supply chain risk. The proposed approaches are based on the experience of subject matter experts professionally engaged in risk

assessment. In this respect, AHP methodologies decompose the decision making problem into sub-problems that are assigned different weights reflecting decision makers preferences. Then, the alternatives are ranked by a sequence of pair-wise comparisons.

The paper by Chand *et al.* [32] highlights key areas of supply chain related risk factors, including those specific to operations and transportation. It provides a framework that allows to proactively manage supply chain risk and to conduct risk assessment in the supply chain to find out the best supply alternatives. In this respect, it leverages analytic network process (ANP) as an extension of the AHP method along with multi-objective optimization in order to derive the order of preference for decision making. The ANP method can factor interdependence among decision making elements in order to obtain composite weights. Subsequently, these weights can be used to rank alternatives in a similar manner to a Markov chain process but without the ability to assess requirements captured as formal verification properties.

2.3 Information Sharing for Plan Execution Monitoring

Plan execution monitoring represents a complex yet essential process in the context of an active, risk-prone execution environment. For plan related logistics activities in particular, an important aspect is related to the completion of deliveries along with the anticipation of the need to initiate contingencies or replanning in order to adapt to adverse conditions. Alongside, correction of plan deviation represents a key aspect in uncertain environments as encountered during humanitarian aid distribution, situations of crisis, rescue operations, etc. Such efforts usually involves the participation of multiple partners, which need to have shared information awareness as described by Allen *et al.* [8]. However, from a strategic level organizational perspective, decision makers may prefer partial information sharing with other participants as this may involve different policy or security issues. Therefore, a distributed solution might need to also include such aspects. In the context of plan execution monitoring and information sharing among collaborating plan participants, decentralization allows distributed parties to pursue specific goals according to their capabilities while aiming towards effective information sharing.

Collaborative and distributed monitoring may also involve the modelling of the plan execution monitoring processes with respect to information access, analysis procedures and information

sharing mechanisms. Another important aspect in this regard, is represented by the employed protocol, which represents a key factor for appropriate data gathering and information exchange among distributed participants.

For many organizations, the established hierarchy for decision making can be reflected in a tree structure. In such structure, each level needs to receive information from the level below in order to analyse and extract the meaningful information to be provided to the level above.

In the scope of interest of this thesis, the focus is on monitoring the execution of operational plans in relation to the events generated during the execution of plan related tasks. Typical such tasks include logistic support, which usually involves the delivery of goods with minimized cost through potentially changing environment conditions. We are interested in the identification of specific needs of gathering and sharing information from multiple plan executing participants that can aggregate into communicating clusters. In this context, an advanced plan execution monitoring framework can greatly benefit from having a formal foundation, which can enable rigorous analysis on various properties of interest that can be formally captured. In this pursuit, this thesis includes the elaboration of an information sharing mechanism with lightweight gossip-like communication capabilities at the cluster level and asymmetrical neighbouring relationships across hierarchical levels. In this regard, we favour the clustering of participants whereby the gossip based interaction is limited to the neighbourhood of a defined clustering distance [57]. In the following, we discuss various relevant works in our scope of interest, of which the most noteworthy ones are categorized as depicted in Figure 2.2 and Table 2.2.

Area of interest \ Approach or technique	Data Aggregation Simulation	Data Exchange Simulation	Probabilistic Formal Assessment	Symbolic Formal Assessment	Network-oriented Assessment	Analytic Assessment
Tree-based protocols	[41, 92]					
Random-walk protocols	[14]					
Gossip-based and/or Flooding-based protocols	[4, 77]	[122]	[53, 84]			
Resource discovery protocol	[135]					
Plan-related group learning and/or emergency plan data sharing	[31, 115]					
Organizational supply management	[143]					

Table 2.2: Noteworthy approaches or techniques in the interest area of “Information Sharing for Plan Execution Monitoring”

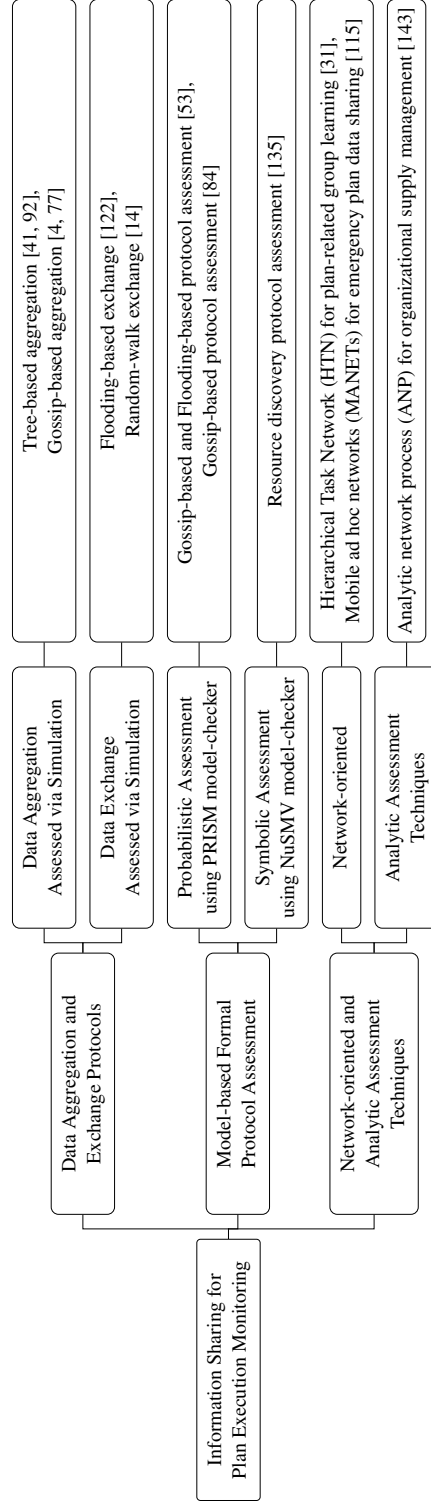


Figure 2.2: Categorization of noteworthy initiatives in the scope of “Information Sharing for Plan Execution Monitoring”

2.3.1 Data Aggregation and Exchange Protocols

Tree-based protocols are discussed by Dam and Stadler [41] as well as by Madden *et al.* [92] where nodes are organized into a spanning tree. Both of these works use simulation in order to exercise the protocols and to characterize their findings. In essence, the leaf nodes send updates of their local variables to their corresponding parents in the hierarchy. In turn, all parents compute partial aggregates based on the updates received from their respective children. This way, the global aggregate is eventually formed at the root node. However, an up-to-date situational awareness may also require the exchange of information among peers on the same hierarchical level. This is especially noteworthy at the tactical level where positional awareness and trafficability information has to be exchanged among the participants.

Various protocols for distributed computation of monitoring aggregates are available in the literature [4, 14, 77, 122]. Flooding-based protocols [122] initiate a network flood whereby all nodes are updated. However, this is not scalable, as the load on every node is rapidly increasing with system size. Random-walk protocols [14] propagate the node state information to a randomly selected neighbour, which in turn updates and further relays the information to yet another randomly selected neighbour. Consequently, the information eventually propagates to all the nodes. However, scalable state representation is an inherent issue for this procedure. Gossip (epidemic) aggregation protocols [77, 100] have also been proposed for randomized communication. The key idea is based on the property that each node is holding an estimate of the aggregate that is exponentially converging to the global aggregate.

2.3.2 Model-based Formal Protocol Assessment

Formal models for flooding and gossiping are explored by Ansgar and Peng [53] with respect to the choice of modelling and resulting performance. This work contrast probabilistic model-checking analysis against simulation-based assessment. PRISM model-checker is used to carry out formal analysis on small scale networks while simulation is used for large networks. Model-checking analysis is used to generate assessment results that cannot be obtained via simulation, which uses statistics and approximations. With respect to assessing gossip-based protocols using probabilistic model-checking, PRISM is used by Kwiatkowska *et al.* [84] in order to analyse

a gossip-based protocol for propagating network topology changes among the nodes of a communication network. In this work, each node is passing information to a small, random subset of other nodes, each of which further passes the information in a similar manner. However, this work does not consider a hierarchy among nodes nor faulty communication due to disruptive environment. The challenge related to finding resources or services in grid computing environments is addressed by Alireza and Jafari [135] using formal verification as alternative to simulation-based approaches. Grid computing involves collaborative sharing of computing resources from multiple networked nodes. To this end, the authors propose automatic verification for resource discovery via symbolic model-checking. In this respect, properties of control behaviour are captured as CTL (Computation Tree Logic) and LTL (Linear Time Logic) specifications in order to formally verify data gathering and data discovery behaviours. In this context, the NuSMV [34] model-checker is employed to discover multi-attribute and range queries and to detect logical inconsistencies.

2.3.3 Network-oriented and Analytic Assessment Techniques

A Hierarchical Task Network (HTN) planning approach is proposed by Chalco *et al.* [31] in order to automate and optimize tasks for instructional designers in Computer Supported Collaborative Learning (CSCL). The paper employs a model whereby strategies are defined as hierarchical tasks and methods while the underlying CSCL scenario is considered as a planning problem. In this setting, CSCL scenarios involve units of learning that are organized in order to support group learning processes.

In the context of mobile ad hoc networks (MANETs), the behaviour of routing protocols is analysed by Quispe and Galan [115] for emergency plan scenarios in urban area. This work discusses proactive, reactive and hierarchical strategies for MANETs, which represent collections of mobile nodes that can communicate in wireless manner and dynamically establish a network without fixed infrastructure. This is useful in emergency situations requiring quick information sharing. The NS2 [73] simulator was used in the study in order to show that hierarchical routing is better suited as it partitions the network into subsets of nodes, thereby localizing data exchange while streamlining data aggregation and forwarding, as part of hierarchical routing strategy.

An analysis of green supply chain management (GSCM) under uncertainty is carried out by Ming-Lang *et al.* [143] with respect to close-loop and open hierarchical structures used as part

of an ANP-based approach. Moreover, the study uses ANP method in conjunction with fuzzy sets. The results of the aforementioned work show that a close-loop hierarchical structure is more appropriate for operational processes and the underlying management activities. However, ANP method is based on subjective weights reflecting decision makers' preferences. This contrasts to probabilistic modelling of hierarchical interactions among asymmetrically clustered participants, which allows to formally assess properties, expressed in probabilistic temporal logic, via probabilistic model-checking.

2.4 PRISM Model-Checker and Markov Decision Processes

The logistic-related risk assessment and plan execution monitoring contributions of this thesis involve the use of model checking-based probabilistic formal assessment techniques. In this respect the selected probabilistic model-checker is PRISM [65]. The latter represent a powerful and well known probabilistic model-checker that is frequently used in the research community. As such, it has been used to assess systems from various application domains, including communication networks [1], distributed systems [7], security protocols [98], etc. We detail next the key aspects and capabilities of PRISM (probabilistic symbolic model-checker). It was initially developed at the University of Birmingham and then enhanced at the University of Oxford. It represents a powerful tool for formal modelling and analysis of systems exhibiting stochastic or probabilistic behaviour. PRISM uses efficient data structures and the numerical methods that it applies are both time and memory effective.

Moreover, PRISM allows determining the actual probability value for a given behaviour occurrence in the analysed model as well as checking the satisfaction of probabilistic properties. These properties usually capture functional requirements and performance specifications of the system. PRISM supports different types of probabilistic models:

- Discrete-Time Markov Chains - DTMC [2].
- Continuous-Time Markov Chains - CTMC [21].
- Markov Decision Process - MDP [114].

In the scope of interest of the thesis, MDP represents a suitable model for capturing decision making under uncertainty. MDP represents a compact discrete-time transition systems with discrete probability distributions that allows for nondeterministic choices. In essence, MDP extends DTMC with the ability to have multiple transition choices in each state in addition to the transition dynamics of DTMC where transitions only occur in every state based on a defined probability distribution among possible transitions.

An MDP can be succinctly described by the tuple $\langle S, A_s, P(s, s'), R(s, s') \rangle$ where:

- S represents a finite set of states.
- A_s represents a finite set of actions available in state $s \in S$.
- $P_a \in P(s, s')$ represents the probability of action a in state s to cause a transition to state $s' \in S$.

In addition, rewards can be employed if needed via predicates that determine reward conditions. In this respect, $R_a \in R(s, s')$ can be specified as reward for making the transition from s to s' . The language used by PRISM makes use of reactive modules, which can be composed in parallel. The modules contain variables, commands labelled with actions for synchronization, guards as well as probabilities. A PRISM model specifies one or more modules, which may interact as a result of variable value changes or via action synchronization. Each module specifies local variables such that at any given time, the values of the specified variables constitute the individual state of the module. In turn, the global state of the whole model is determined by the states of all specified modules. The dynamics of each module is described by a set of commands (potential actions in A_s), usually taking the form:

```
[action] guard -> probability_1: (var_1'=update_1)+
                    probability_2: (var_2'=update_2)+
                    ...
                    probability_n: (var_n'=update_n);
```

The `[action]` label is used to indicate that the command is synchronized with other command(s) labelled with the same action. Thus, this feature can be used to capture communication.

However, a command for which an action label is not specified can be evaluated without the need to synchronize with other commands. The guard represents a predicate that can span over all variables in every module of the model. A fired transition results in updating the values of one or more variables in the module, possibly as a function of other variables from the whole model.

The listed updates can be assigned probabilities corresponding to their transitions such that the sum of the probabilities corresponding to each of the updates in a command must sum to unity:

$$\begin{aligned} &probability_k \in P(s, s'), k = 1..n, \text{ such that} \\ &\sum_{k=1}^n probability_k = 1. \end{aligned}$$

In addition, quantitative rewards can be specified if needed in order to mark the satisfaction of corresponding predicates:

$$reward_k \in R(s, s'), k = 1..m.$$

The predicates are evaluated at each step, allowing to accumulate the rewards:

```
rewards
    reward_predicate_1: reward_1;
    reward_predicate_2: reward_2;
    ...
    reward_predicate_m: reward_m;
endrewards
```

PRISM is equipped with property specification languages. These include the Continuous Stochastic Logic - CSL [17] with applicability to the CTMC model and the Probabilistic Computation Tree Logic - PCTL [50], which is applicable to DTMC and MDP models [63]. PCTL represents an extension of Computation Tree Logic - CTL [18] mainly with probability operator. Since MDPs are employed in the corresponding contributions of the thesis, the required properties are expressed in PCTL. The syntax of PCTL is provided below in brief:

$\phi :=$	$true$	
	a	(Atomic proposition)
	$\neg\phi$	(Negation)
	$\phi \wedge \phi$	(And)
	$\mathcal{P}_{\bowtie p}[\psi]$	(ψ is true with probability $\bowtie p$)
$\psi :=$	$\phi_1 \mathcal{U}^t \phi_2$	(Bounded until)
	$\phi_1 \mathcal{U} \phi_2$	(Until)
	$X\phi$	(Next)
$t \in$	N ;	$p \in [0, 1]$
$\bowtie \in$	$\{>, \geq, <, \leq\}$.	

In PCTL one can also utilize the finally (F) operator given by the PCTL equivalence:

$$\mathcal{P}_{\bowtie p}[true \mathcal{U} \phi] \equiv \mathcal{P}_{\bowtie p}[\mathcal{F} \phi].$$

However, the until (U) operator is general and required as part of PCTL syntax definition while the (F) operator provides syntactic sugar to improve the readability of more elaborated properties.

2.5 Asset Localization and Sensor Networks

Various categories of organizational assets involved in the execution of plan specific tasks (e.g., logistic support) may have self-localization abilities. Typical such assets include those belonging to the same organization (e.g., self-reporting using on-board GPS). In contrast, others categories of assets may require external sensors in order to be localized (e.g., organizational assets without on-board GPS or assets belonging to other organizations).

Moreover, asset localization problems also have a characteristic importance for surveillance and rescue efforts as mentioned by Sonia and Niki [146]. In this respect, asset localization and related problems have been studied as well in connection to wireless sensor networks (WSNs). We subsequently discuss several relevant works in the scope of interest of the thesis, of which the most noteworthy are categorized as illustrated in Figure 2.3 and Table 2.3.

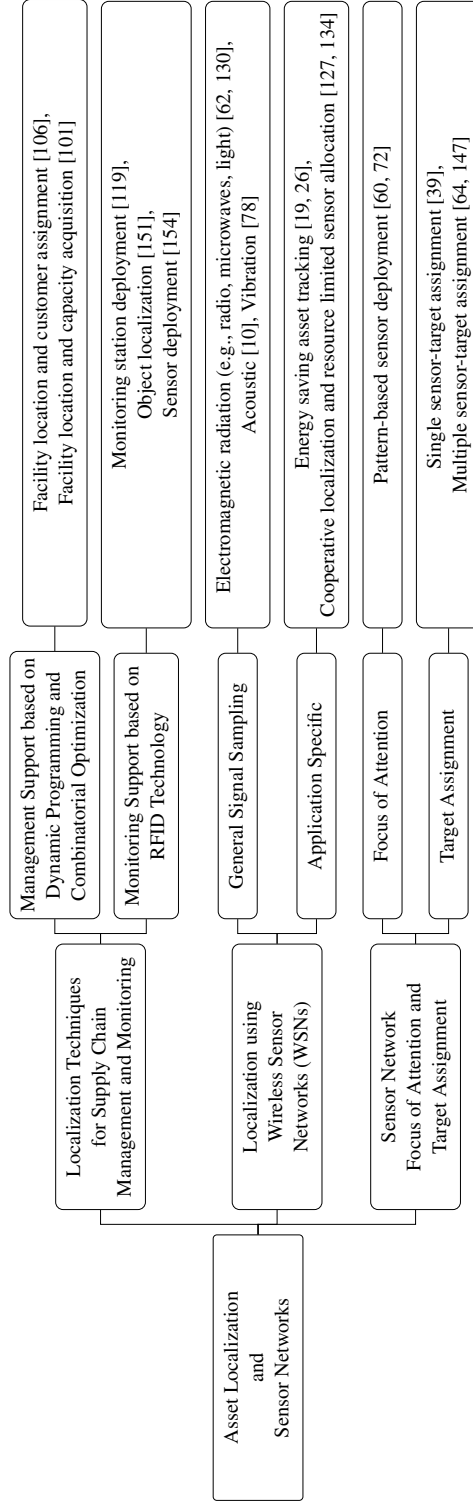


Figure 2.3: Categorization of noteworthy initiatives in the scope of “Asset Localization and Sensor Networks”

Area of interest \ Approach or technique \	Simulation and/or Dynamic Programming	Analytic Combinatorial Optimization	RFID Technology	Signal Sampling	Bayesian and/or Case-based Reasoning	Heuristics and/or Simulation
Supply chain support for management	[101]	[106]				
Supply chain support for monitoring			[119, 151, 154]			[119]
General localization using WSNs				[62, 130] [10, 78]		
Resource constrained localization using WSNs	[26]		[26]		[127, 134]	[19]
Sensor focus of attention and/or target assignment	[39]	[60, 72, 147]				[64]

Table 2.3: Noteworthy approaches or techniques in the interest area of “Asset Localization and Sensor Networks”

2.5.1 Localization Techniques for Supply Chain Management and Monitoring

Localization problems are encountered in supply chain management, as discussed by Özsoy and Pinar [106]. This work considers locating facilities with limited capacity while performing corresponding customer assignments that minimize the maximum distance between customers and the facility that they are assigned to. The aforementioned work divides the initial problem into sub-problems that are iteratively solved using an exact algorithm based on combinatorial optimization. The proposed approach is evaluated experimentally. Facility location is also discussed by Murat *et al.* [101], which propose an exact algorithm for the facility location and capacity acquisition problem where a number of facilities need to be located on a service line. This work employs dynamic programming to prioritize allocation decisions while conducting a computational study on the effects of demand density on the generated solutions. More details on facility location are provided in Section 2.6.1 in the context of logistic support.

In the area of supply chain management and monitoring, the work of Ray *et al.* [119] addresses the problem of Radio Frequency IDentification (RFID) enabled monitoring station deployment. In this work, a heuristic technique is employed whereby multiple participants collaborate in order to deploy monitors such that the energy consumption of the related monitoring relay nodes is minimized. Benchmark results are also provided. In relation to object localization, Yang *et al.* [151] propose an improved object localization approach that employs a sparse RFID tag distribution. In this area, there are also specific applications such as sensor deployment, as indicated by Zhao and Gan [154]. The forenamed work considers a distributed design of RFID sensors for large-scale network deployment.

2.5.2 Localization using Wireless Sensor Networks

Target localization represents a key application [137] for WSNs, which are composed of multiple nodes that are deployed randomly or strategically (vantage points, junction points, etc.), as detailed by Boutaba *et al.* [27]. Each WSN node is typically resource constrained [102] and able to detect targets in its range by sampling signals such as electromagnetic radiation (e.g., radio-waves, microwaves, light), sound, etc., as mentioned by Gustafsson and Gunnarsson [62]. Multiple sensors of various capabilities can provide sensor diversity, which typically enhances target localization, as noted by Souza *et al.* [136]. A radar-based sensor network is proposed by Sobhani *et al.* [130] for detecting targets via ultra-wideband radio impulse. Acoustic sensors have been studied by An *et al.* [10] for their potential to distinguish objects that have different sound signatures. Vibration sensors are considered by Jin *et al.* [78] in order to detect different types of objects. In this work, seismic and passive infrared sensors are employed to detect and classify humans, animals and light vehicles. An interesting architecture aimed at tracking assets within construction sites is discussed by Bisio *et al.* [26]. The employed components are cost effective and involve Radio Frequency IDentification (RFID), Bluetooth Low Energy (BLE) tags in conjunction with mobile devices such as smart phones. The core functionality is provided by a pair of applications implementing asset searching and tracking. The benefits of the proposed architecture reside in the ability to extend mobile device battery lifetime while providing good localization accuracy. In addition, the architecture allows for a trade-off between energy consumption and localization accuracy while the effectiveness of the proposed solution is demonstrated via simulation.

Furthermore, Balakrishnan and Nayak [19] propose an efficient geographic asset tracking solution that allows conserving mobile resources such as energy used for communication, by dynamically adapting the tracking scheme via context-aware personalized route learning. This involves distributed proactive monitoring of context information regarding asset specific properties (from its routes characteristics), collected under different environmental conditions. The proposed approach allows for an optimized evaluation of data transmission with a reduced overhead, which results from personalizing the tracking algorithm for the mobile assets.

The problem of cooperative localization in mobile networks using belief propagation is addressed by Savic and Zazo [127] in the context of mobile networks. The paper discusses the main issues specific for this type of problem, such as the high communication cost and the sampling

techniques' challenges. A multi-agent system involving negotiating agents pursuing resources allocation activities is presented by Soh and Tsatsoulis [134]. This work is addressing specific issues of constraint satisfaction in multi-sensor target tracking. The agents aim to optimize the use of their own consumable resources while pursuing the global goal of multi-sensor target tracking. In addition, the agents can use different strategies selected using case-based reasoning.

2.5.3 Sensor Network Focus of Attention and Target Assignment

The focus of attention problem (FOA) is considered by Goossens *et al.* [60] over a sensor network consisting of small, simple but cost effective sensors that are used for estimating target locations. The sensors are arranged on a straight line while the targets are situated somewhere in the plane. The objective of the underlying combinatorial optimization problem is to assign sensors to targets such that the overall expected error of the target locations is minimized. In this setting, the paper considers the special case where every target is tracked by a pair of sensors, typically representing cameras. The authors highlight that to use WSNs effectively, specific challenges must be addressed. The challenges are typically related to the limitations of individual sensors since in general, a single sensor cannot estimate the location of a target. In this context, an important challenge relates to the need to have multiple sensors per target and to mitigate interfering effects such as the presence of noise. In this setting, a key aspect is related to which sensors need to be assigned to which targets and how to combine the measurements in order to get a reasonably accurate estimation. The work presented by Isler *et al.* [72] addresses a similar problem where multiple range sensors can be placed in a circular arrangement for monitoring a particular region.

Specific assignment problems arising from multiple target tracking are discussed by Poore and Gadaleta [113] in relation to data association and sensor set partitioning. Moreover, issues relating to sensor coverage are discussed by Cortes *et al.* [39] where a single sensor is considered as sufficient to cover a particular target but the coverage quality decreases with distance. This work aims at ensuring optimal coverage over sensor network movements. Furthermore, different assignment problem flavours are discussed in the literature by Pentico [111] and by Burkard *et al.* [29]. In this respect, the Multidimensional Assignment Problem [112], also termed as the axial Multi Index Assignment Problem [20] represents a well-known optimization problem. Walteros *et al.* [147] describe a variant of the multidimensional assignment problem in the context of multi-sensor

multi-target tracking problems, where different sensor measurements from a sensor network must be matched to different targets. The solution approach involves mixed integer linear optimization and the use of branch and price algorithm. A branch and bound algorithm for the multidimensional assignment problem is presented by Larsen [85] along with local search improvements. Also, a decomposition scheme is proposed by Vogiatzis *et al.* [145] for partitioning the multidimensional assignment problem in a manner that allows for subsequent recombination. This can provide upper and lower bounds for the original problem. Moreover, vehicle-target assignment is studied by Arslan *et al.* [12] using game theory whereby a group of vehicles aim to optimally assign themselves to a given set of targets. The paper presents simulations illustrating vehicle negotiations that can lead to near-optimal assignments. Furthermore, a memetic algorithm [64] combining a genetic algorithm with a form of local search is proposed by Karapetyan and Gutin [81] for a variation of the multidimensional assignment problem. In this setting, the forenamed work is employing an adjustable population size to improve the efficiency of the local search procedure.

2.6 Logistic Support Planning and Risk Mitigation

Logistics management is well studied in the scientific community. In general, logistic support activity planning for operational plans corresponds to supplying resources in a timely and cost effective manner at the appropriate demand nodes. Logistic support activities involve specific operational research problems that have characteristic NP-Hard complexity. These include network partitioning, as discussed by Feder *et al.* [52] and by Oncan *et al.* [139], clustering, as discussed by Kanungo *et al.* [80], facility location, as detailed by Farahani *et al.* [51] and by Azarmand and Neishabouri [16], along with various vehicle routing problems (VRP). For the latter, a taxonomy is provided by Braekers *et al.* [28]. Due to combinatorial explosion, such problems have in general huge solution spaces with irregular structure. This renders of little value the attempts to prune significant parts of the solution space, looking for the optimal solution, which requires visiting the complete solution space in the worst case and a significant fraction thereof on average. As such, this hints to the need of employing heuristics in pursuit of finding near-optimal solutions in a computationally tractable manner. Next, we discuss a number of relevant works in our scope of interest, of which the most noteworthy are categorized as presented in Figure 2.4 and Table 2.4.

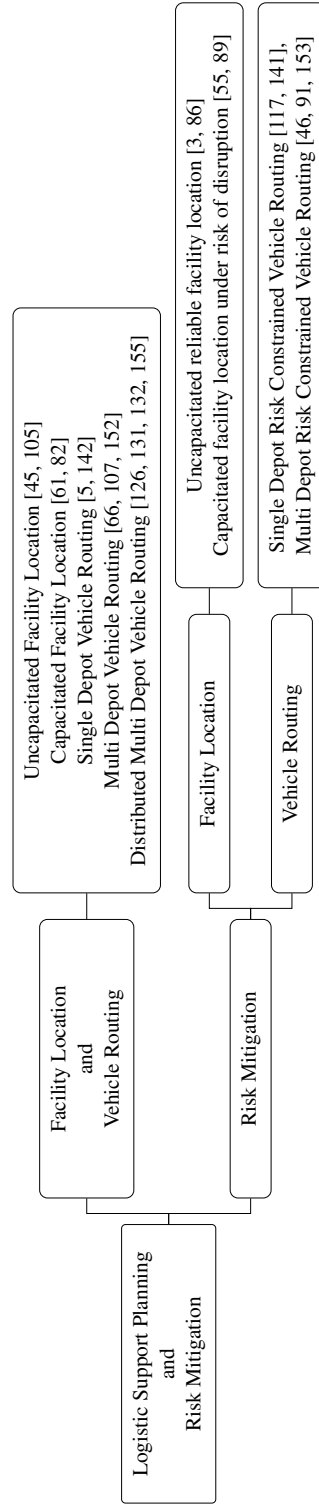


Figure 2.4: Categorization of noteworthy initiatives in the scope of “Logistic Support Planning and Risk Mitigation”

Area of interest \ Approach or technique \	Evolutionary and/or Learning Heuristics	Analytic Combinatorial Optimization	Game Theory	Ant Colony and/or Genetic Algorithm	Bayesian Networks and/or Fuzzy Programming	Tabu and/or GRASP
Capacitated/uncapacitated facility location	[61, 105]				[45, 82]	
Capacitated/uncapacitated facility location with risk mitigation	[3]	[55, 89]				[86]
Single depot vehicle routing		[142]		[5]		
Single depot vehicle routing with risk mitigation				[141]		[117]
Multi depot vehicle routing		[107]		[66, 152]		
Distributed multi depot vehicle routing	[131, 132]		[126, 155]			
Multi-depot vehicle routing with risk mitigation		[153]		[91]	[46]	

Table 2.4: Noteworthy approaches or techniques in the interest area of “Logistic Support Planning and Risk Mitigation”

2.6.1 Facility Location and Vehicle Routing

Facility location problem (FLP) and vehicle routing problem (VRP) are very important for supply chain and logistic support planning. In essence, FLP aims at minimizing the combined facility establishment and direct transportation cost to meet customer demands while VRP aims at finding the routes with minimum cost for a fleet of vehicles, in order to serve customer demands. FLP is considered in either uncapacitated or capacitated settings. In the former case, any number of customers can be assigned to a facility whereas in the latter case, a limited number of customers can be assigned to a facility depending on its capacity. An approximation algorithm for FLP is presented by Li [87] while Dogan [45] details an approach based on Bayesian networks. Moreover, an approach leveraging evolutionary computing is utilized by Otto and Kokai [105]. For the capacitated FLP, a hybrid evolutionary learning algorithm is proposed by Guo *et al.* [61] while Tarik *et al.* [82] employ fuzzy programming.

With respect to VRP, Alba and Dorronsoro [5] detail a very effective cellular genetic algorithm. Also, various modelling approaches and related solution generation techniques are presented by Toth and Vigo [142] and by Golden *et al.* [59] for VRP and its different variants such as: VRP with time-windows (VRPTW), distance constrained VRP (DCVRP), VRP with pickup and delivery (VRPPD), etc. For multi depot VRP (MDVRP), there are specific heuristics and meta-heuristics solution generation techniques. These include ant colony optimization as detailed by Yu *et al.* [152] as well as genetic algorithms as described by Ho *et al.* [66], etc. Such techniques employ populations of solutions that evolve based on specific fitness criteria in order to form better

solutions. Also, for MDVRP with split-delivery, Ray *et al.* [120] detail a learning-based generative heuristic based on the work in [132]. In addition, Zibaei *et al.* [155] propose for MDVRP a distributed solution approach based on cooperative game theory. Furthermore, a multi-stage approach employing Lagrangian relaxation is discussed by Özyurt and Aksen [107]. The proposed approach involves the initial partitioning of the transport network, followed by facility location and vehicle routing. Solution generation for MDVRP in decentralized or distributed manner requires the participants to share in full [132] or in part [131] the problem data and the obtained results. In this setting, the participants are solving vehicle routing problems for which they may obtain different near-optimal results (partial or complete solutions) depending on their employed solution generation techniques and/or available computing resources. Result sharing is also instrumental in order to reduce the computational load. Collaborative solution generation for MDVRP is approached in [132], using a learning-based heuristic with full information sharing while in [131], evolutionary learning is employed with partial information sharing. When the participants have restrictions in fully sharing the problem input (e.g., due to security concerns, organization policy), distributed solution techniques based on competitive game theory [126] can also be employed.

2.6.2 Risk Mitigation

A model for reliable logistics network under potential facility disruption is provided by Peng *et al.* in [110]. This work aims at cost minimization while reducing the disruption risk via cost bounding. Moreover, Shen *et al.* [128] present a heuristic algorithm for the reliable facility location problem where there is a risk that some facilities may occasionally fail. For the capacitated FLP where some facilities are exposed to risk of failure, Gade and Pohl [55] detail the concept of redundant facility along with a related mathematical model. Li and Savachkin [86] develop a Tabu search heuristic for the reliable facility location problem where facilities have heterogeneous failure probabilities while a finite budget can be used for facility fortification. Another approach for this type of problem is provided by Afify *et al.* [3] using an evolutionary learning algorithm. Also, Liberatore *et al.* [89] propose a model for optimizing protection plans with limited resources for capacitated systems that are exposed to partial or complete disruption.

The concept of risk on route is typically encountered in relation to the transportation of chemical and hazardous materials (hazmat), as discussed by Bianco *et al.* [25]. In this respect,

various models have been developed where a risk function is employed for each considered road segment. Then, routes with higher safety levels are subsequently selected in order to minimize the underlying operating costs. The risk functions factor specific aspects such as the material being transported along with the road characteristics, as mentioned by Van Raemdonck *et al.* [118].

Talarico *et al.* [140] propose a metaheuristics for the risk-constrained cash-in-transit vehicle routing problem (RCTVRP) along with a subsequent ant colony extension [141]. In essence, RCTVRP introduces a maximum risk threshold constraint over VRP routes to limit the risk for cash pickup operations where robbery is a concern. Each pickup location must be visited only once while the number of vehicles is not limited. The risk of a vehicle is considered as proportional to both the amount of cash carried as well as the time or the related distance covered by the vehicle. In addition, vehicle capacity is not considered relevant for RCTVRP. However, a single vehicle solution is prohibited in general as it would exceed the maximum risk threshold. Furthermore, Radojicic *et al.* [117] propose a Greedy Randomized Adaptive Search Procedure (GRASP) for RCTVRP, hybridized with path re-linking methodology. Path re-linking enhances GRASP via an intensification strategy that allows the exploration of trajectories that connect GRASP solutions and the best elite solutions previously encountered.

Zhao and Zhu [153] consider the problem of explosive waste recycling during transportation among multiple depots. The authors develop a multi-depot vehicle routing model for the minimization of total cost and total risk, by planning vehicle tours for explosive waste collection along with return trips between collection centers and recycling centers.

Ma *et al.* [91] consider the multi-depot vehicle routing problem (MDVRP) for hazardous materials. The paper provides a multi-objective optimization model for minimizing the total transportation energy consumption and transportation risk. It also proposes a two-stage solution method (TSM) and a hybrid multi-objective genetic algorithm (HMOGA). The TSM allows determining the customers to be assigned to the depots using a clustering technique that considers energy consumption, transportation risk, and depot capacity in the first stage. This is followed by determining the servicing order using a multi-objective genetic algorithm in the second stage. In the case of HMOGA, the assignment of customers to the depots and the customer service order are optimized simultaneously. As such, while being more complex, HMOGA generally outperforms TSM. Also, Du *et al.* [46] propose a fuzzy bi-level (upper/lower) programming model for minimizing the total

expected transportation risk when delivering hazardous materials from multiple depots. The upper level allocates the customers to the depots by observing depot capacities and customer demands. Then, the lower level provides the minimized path for each group of depot and customers. Fuzzy simulation-based heuristic algorithms are also detailed along with illustrative examples.

2.7 Summary

This chapter provided an overview of previous research initiatives relevant for the topics studied in this thesis. In this context, we first discussed related techniques that are useful for plan execution and monitoring. Then, we discussed plan-related logistic support risk assessment and associated monitoring and plan adaptation activities. Thereafter, we reviewed a number of noteworthy research directions on plan execution monitoring in collaborative setting. Subsequently, important details have been presented on the probabilistic model-checker PRISM and its benefits, given that two of the contributions of this thesis employ probabilistic model-checking. After that we discussed specific initiatives related to asset localization and concluded the chapter with a discussion on logistic planning and risk mitigation.

Chapter 3

Risk Analysis Using Probabilistic Model-Checking

Most planning activities involve logistic support and related supply chain activities, which represent essential components in many endeavours covering both public and private domains. In this respect, a key capability area of interest relates to obtaining enhanced decision support based on conducting risk analysis on the logistic support components of organizational plans. Such capability is instrumental given that transport networks are complex and potentially fragile due to weather, natural disasters or other risk factors. Thus, assessing transportation related risk represents a key decision support capability along with the ability to evaluate contingency options for risk mitigation.

In this chapter, the aforementioned issues are addressed by adopting probabilistic model-checking to evaluate risk exposure and plan related contingency options, which are illustrated in the context of logistic support planning. In this pursuit, risk related properties are assessed using the MDP probabilistic behavioural model to capture the transport system. Moreover, we show the usefulness of constructing decision trees that can provide insightful means of risk appraisal. The proposed approach can help decision makers evaluate contingency options and determine lower and upper cost bounds for risky transportation tasks such as those involved in humanitarian aid provision. We provide next important background information on the use of decision trees as well as on the use of the selected probabilistic model-checker that is employed.

3.1 Decision Trees

Decision trees provide a graphic representation for comparing decision alternatives and allow to assign values to those alternatives by combining uncertainties, cost values, and payoffs. In the usual notation, squares represent the decisions to be made. The branches stemming from a square correspond to the choices available to the decision maker. Circles represent chance nodes (probabilistic events). The branches stemming from a circle represent the possible outcomes of a chance event. We discuss in the following an illustrative example from [36]. Suppose that person P1 has a ticket for a game of chance that will provide \$10 with a 45% chance, and nothing with a 55% chance. Also, person P2 has a ticket to a different lottery with 20% chance of gaining \$25 and an 80% chance of gaining nothing. Then, consider that P2 proposes a ticket swap to P1 but asks one dollar for the exchange. The decision making question is: should P1 agree to do the swap and give one dollar to P2 in the hope of winning \$25, or should P1 keep his ticket and have a better chance of winning \$10?

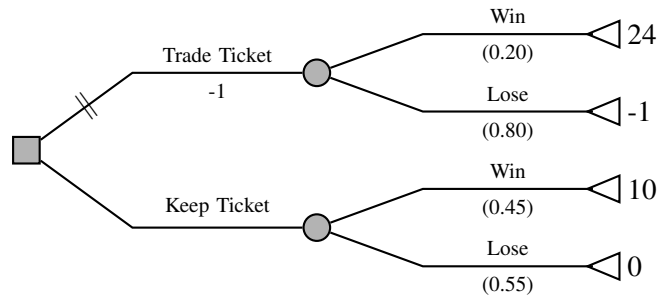


Figure 3.1: Decision tree example

Figure 3.1 depicts the decision tree where the numbers at the end of the branches represent the net values. If P1 swaps tickets and wins, the net amount gained is \$24, having paid one dollar to P2. Using the concept of Expected Monetary Value - EMV from [36], the decision tree can be assessed as follows. First, we calculate the EMV in the case of trading the ticket:

$$\text{EMV}(\text{Trade Ticket}) = 0.20 \times 24 + 0.80 \times (-1) = \$4$$

Then, we calculate the EMV in the case of keeping the ticket:

$$\text{EMV}(\text{Keep Ticket}) = 0.45 \times 10 + 0.55 \times 0 = \$4.5$$

Finally, after replacing the chance nodes in the decision tree with the corresponding expected monetary values, P1 can decide between trading and keeping the ticket by choosing the branch with the highest expected monetary value (Keep Ticket).

3.2 System Description

The considered logistic delivery system includes a number of depots and transport vehicles. The delivery involves routes with associated levels of risk on various route segments, which may lead to vehicle failure. In our scope of interest, MDP represents a suitable model for risk-prone transport system since it can be used to capture multiple vehicle choices over unreliable transport network, allowing for nondeterministic choice.

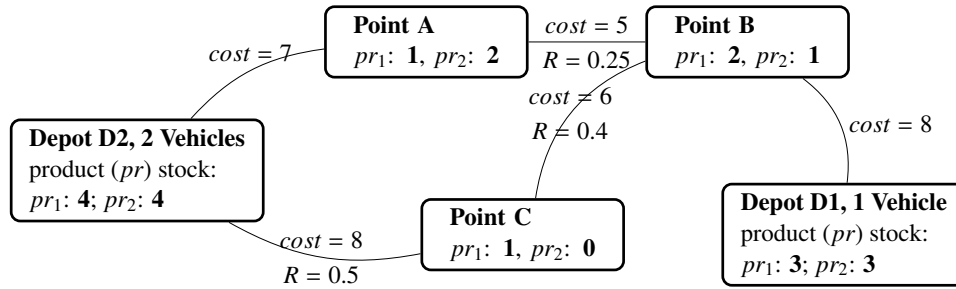


Figure 3.2: Case study

We present the proposed risk and contingency assessment technique using a case study. Thus, Figure 3.2 depicts an example of a transportation system involving 2 depots (D1 and D2) and 3 demand points (A, B and C). Depot D1 can use one vehicle whereas depot D2 can use at most two vehicles. The routing segment costs, the presence of risk (R), the depot stock levels and the demands are annotated in Figure 3.2. The level of stock is listed for illustrative purposes and in order to better convey the approach we abstract it in the sequel.

3.3 Assumptions

In logistics and transportation, risk varies from one route to another especially in adverse conditions (e.g., unpaved or damaged roads, snow-covered roads, tunnels, etc.). The overall risk of a transportation task becomes higher when more vehicles are travelling in adverse conditions.

Route Segment	Success Probability	Failure Probability
$A \Rightarrow B$	0.75	0.25
$B \Rightarrow A$	0.75	0.25
$B \Rightarrow C$	0.6	0.4
$C \Rightarrow A$	0.6	0.4
$D2 \Rightarrow C$	0.5	0.5

Table 3.1: Routing-related risk probabilities

Given that different routing choices can potentially exhibit different risk levels, it is necessary to evaluate and address the corresponding risk levels. In this context, we assume established risk parameters and corresponding acceptability criteria as requirements. An acceptable probability for successful delivery in the context of the uncertainty posed by the risky routes is considered to be at least 50% while an acceptable probability of losing all vehicles is considered at most 15%. Considering Figure 3.2, we assume the routing-related risk probabilities as shown in Table 3.1.

3.4 Approach

We employ PRISM to capture vehicle dynamics over the available transport links and address risk-related issues of transportation via probabilistic model-checking. This approach can provide decision support based on formal techniques in contrast to traditional techniques involving simulation. Of the probabilistic models supported by PRISM, we favour MDP compared to other probabilistic models such as DTMC. The latter is well suited for synchronous probabilistic systems wherein the components are evolving in lock-step. In contrast, MDP can capture processes that have both probabilistic and non-probabilistic features and that may operate asynchronously. Moreover, MDP allows to derive decision trees, which can be very useful for decision making. For instance, decision trees can be used in the context of assessing the risk associated with a supplier network in order to determine the most adequate number of suppliers needed [23].

In the present approach, the focus is on risk assessment and contingency evaluation with respect to transport task completion. In this setting, we specify each vehicle individually as an agent traversing available transport links. Each agent is described as distinct module and we employ the capabilities of PRISM to compose the modules. In this setting, the dynamics is captured by a compact discrete-time transition system with discrete probability distribution and nondeterminism (key MPD features). Such modelling provides a suitable semantic interpretation capturing:

- Coordination of behaviour.
- Action execution synchronization.
- Nondeterminism and choice.
- Probabilistic path selection.

Given that PRISM requires numerical values for building and assessing the input model, we employ a macro label notation to transparently map numerical values to the elements being modelled while keeping the model code readable.

3.5 Static and Dynamic Entities

There are two different types of entities involved in the modelling process, namely static and dynamic. The static entities represent locations that can be reached by the vehicles. We leverage the model-checking concept for service delivery described in [133] to characterize the entities of the transportation system model. Dynamic entities represent the state of the agent modules and the underlying variables associated to the vehicles, which can take different values depending on vehicle movement. In contrast to [76], in this work, transport link flows have an associated risk of traversal and are assigned corresponding probabilities for success/failure at the level of the agent dynamics (part of potential transitions). In this respect, we capture vehicle failure as a specific transition to an abstract location "x" standing for no further evolution and denoting that the vehicle failed (e.g., it had an accident or mechanical breakdown, etc.). The involved static entities are as follows (excerpt):

```
formula d1=0;
formula d2=1;
formula a=2;
formula b=3;
...
formula x=5;
...
```

The previously mentioned declarations indicate that we will refer to the specified numerical values using the declaration formula (e.g., a stands for the numerical value of 2). The vehicle modules are labelled using the prefix "mod_" followed by "V[depot][vehicle id]". For the first vehicle of depot D1, we have the following:

```

module  mod_V11
  V11: [0..5] init d1;
  v11sa: [n..s] init n;
  v11sb: [n..s] init n;
  v11sc: [n..s] init n;

  [step1] V11=d1 & !sb
    -> (V11'=b) & (v11sb'=s);

  [step2] V11=b & !sa
    -> 0.75: (V11'=a) & (v11sa'=s)
    + 0.25: (V11'=x);

  [step2] V11=b & !sc
    -> 0.60: (V11'=c) & (v11sc'=s)
    + 0.40: (V11'=x);

  [] V11=b & V21=x & (V22=c | V22=x) & !sa
    -> 0.75: (V11'=a) & (v11sa'=s)
    + 0.25: (V11'=x);
endmodule

```

We discuss next the key declarations above and their significance:

- `module mod_V11;` represents the name assigned to the module.
- `V11: [0..5] init d1;` represents the state variable corresponding to vehicle 1 of D1 (V11), initialized with the value corresponding to D1;

- [step1] $V11=d1 \ \& \ !sb \rightarrow (V11'=b) \ \& \ (v11sb'=s)$; represents a module state transition whereby the module state variables can change in the next step if its predicate guard is satisfied. That is, if V11 is in D1 and b was not visited, then V11 moves to b and v11sb changes to s (recording that b has been served).
- [step2] $V11=b \ \& \ !sa$
 $\rightarrow 0.75: (V11'=a) \ \& \ (v11sa'=s)$
 $+ 0.25: (V11'=x)$; represents a module state transition whereby the vehicle state variable can change in the next step to either a or x with respectively the probabilities of 0.75 and 0.25. When changing the vehicle state variable to a, v11sa is also changed to s in order to record that a has been served (visited).
- [step2] $V11=b \ \& \ !sc$
 $\rightarrow 0.60: (V11'=c) \ \& \ (v11sc'=s)$
 $+ 0.40: (V11'=x)$; represents a module state transition whereby the vehicle state variable can change in the next step to either c or x with respectively the probabilities of 0.60 and 0.40. When changing to c, v11sc changes to s (recording that c has been visited). This transition and the previous one may have their guard predicates satisfied at the same time and consequently this captures a nondeterministic decision among the possible choices.

Similar considerations apply for the vehicles of D2:

```

module mod_V21
V21: [0..5] init d2;
v21sa: [n..s] init n;
v21sb: [n..s] init n;
v21sc: [n..s] init n;

[step1] V21=d2 & !sa
  -> (V21'=a) & (v21sa'=s);
[step1] V21=d2 & !sc
  -> 0.5: (V21'=c) & (v21sc'=s)
  + 0.5: (V21'=x);

```

```

[step2] V21=a
    -> 0.75: (V21'=b) & (v21sb'=s)
    + 0.25: (V21'=x);

[step2] V21=c & !sb
    -> 0.60: (V21'=b) & (v21sb'=s)
    + 0.40: (V21'=x);

[step2] V21=c & sb -> true;

[] V11=x & V21=b & !sc
    -> 0.60: (V21'=c) & (v21sc'=s)
    + 0.40: (V21'=x);
endmodule

module mod_V22
V22: [0..5] init d2;
v22sa: [n..s] init n;
v22sb: [n..s] init n;
v22sc: [n..s] init n;

[] V21=x & V22=d2 & !sc
    -> 0.5: (V22'=c) & (v22sc'=s)
    + 0.5: (V22'=x);

[] V11=x & V21=c & !sa -> (V22'=a) & (v22sa'=s);

[] V21=x & (V22=c | V22=x) -> true;
endmodule

```


For the second vehicle of depot D2, namely V22 a noteworthy remark is that it starts moving only in the case where the vehicle of depot D2, namely V21 failed ($V21=x$). This allows modelling the contingency option whereby V22 still available at depot 2 could be dispatched in case that V21 has failed.

We employ the concept of reward available in PRISM for counting the occurrence of specific behaviours of interest in the model, namely reaching the demand nodes. Thus, whenever a node is visited by a vehicle, a reward is given:

```
rewards
    sa=true: 1;
    sb=true: 1;
    sc=true: 1;
endrewards
```

3.6 Case Study on Decision Making based on Decision Trees

We employ the capabilities of PRISM to determine the actual probability value for a given behaviour occurrence or outcome. Furthermore, we use the state space computed by PRISM to construct the decision graph of the model. The rectangular nodes show the state of the vehicles (vehicle:node) and the accumulated rewards (in square brackets). This provides the ability to graphically explore the model behaviour in addition to assessing the risk using property specification. In this respect, a decision maker can easily spot the outcomes of interest where all demands have been served (e.g., $V11:c, V12:b, V22:d2[3]$).

Figure 3.3 provides the synopsis of the proposed risk assessment technique. The transport system is modelled into the corresponding MDP input for the probabilistic model-checker PRISM, which provides the risk assessment output. This output is parsed in order to generate the corresponding decision tree. In order to account for cost, we consider the Estimated Monetary Cost (EMC). The latter is similar to the EMV but reflects cost instead of payoff. Consequently, a decision maker would prefer the alternative with a lower value when using EMC. The decision tree with estimated monetary cost (EMC) depicted in Figure 3.4 is obtained in two stages. The first stage consists in parsing the output generated by PRISM in terms of states and transitions. Then,

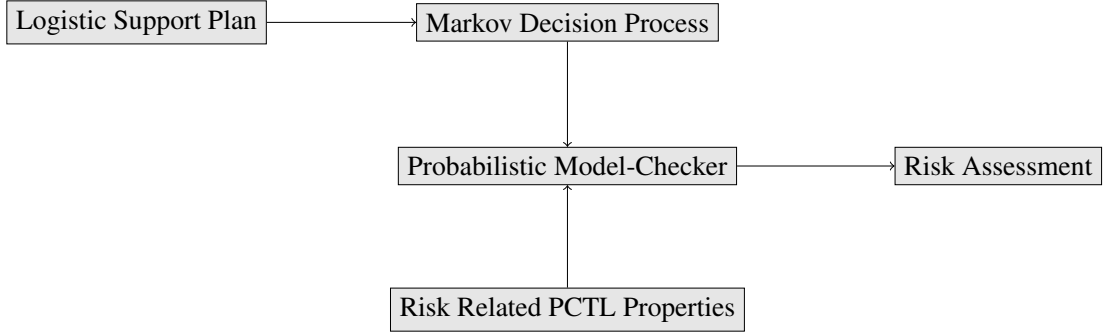


Figure 3.3: Risk assessment technique

the EMC annotation is added at the second stage by computing the cost incurred by the vehicles while moving along the paths corresponding to the decision tree traces. In this context, whenever a vehicle fails, a penalty of 35 is added. The yellow ovals represent chance nodes while the cyan rectangles represent outcomes. All outcomes have corresponding reward counts as well as associated penalty levels consisting in multiples of 35 (with no penalty corresponding to $0 \times 35 = 0$). From the obtained decision tree with EMC annotation, we note that the first decision possibility has better (lower in this case) EMC:

$$EMC_1 = 77 \times 0.5 + 30.25 \times 0.5 = 53.625$$

compared to the second one

$$EMC_2 = 106.5 \times 0.1 + 81.2 \times 0.3 + 56 \times 0.15 + 26 \times 0.45 = 55.11.$$

The rewards corresponding to the possible outcomes of the first decision are as follows:

- 3, with a combined probability given by:

$$\underbrace{(0.5 \times 0.5 \times 0.75)}_{0.1875 \text{ (penalty 35)}} + \underbrace{(0.5 \times 0.25)}_{0.125 \text{ (penalty 35)}} + \underbrace{(0.5 \times 0.75)}_{0.375 \text{ (penalty 0)}} = 0.6875;$$

- 2, with a combined probability given by: $\underbrace{(0.5 \times 0.5 \times 0.75)}_{0.1875 \text{ (penalty } 2 \times 35)}} + \underbrace{(0.5 \times 0.5 \times 0.25)}_{0.0625 \text{ (penalty } 2 \times 35)}} = 0.25$ and

- 1, with a probability given by: $\underbrace{(0.5 \times 0.5 \times 0.25)}_{\text{(penalty } 3 \times 35)}} = 0.0625.$

Moreover, the rewards corresponding to the possible outcomes of the second decision are:

- 3, with a combined probability given by:

$$\underbrace{(0.1 \times 0.5)}_{0.05 \text{ (penalty } 2 \times 35)}} + \underbrace{(0.3 \times 0.4 \times 0.5)}_{0.06 \text{ (penalty } 2 \times 35)}} + \underbrace{(0.3 \times 0.6)}_{0.18 \text{ (penalty 35)}} + \underbrace{0.15}_{\text{(penalty 35)}} + \underbrace{0.45}_{\text{(penalty 0)}} = 0.89 \text{ and}$$

- 2, with a combined probability given by: $\underbrace{(0.1 \times 0.5)}_{0.05 \text{ (penalty } 3 \times 35)} + \underbrace{(0.3 \times 0.4 \times 0.5)}_{0.06 \text{ (penalty } 3 \times 35)} = 0.11.$

For the first decision possibility, the outcomes provide an overall expected reward given by: $3 \times 0.6875 + 2 \times 0.25 + 1 \times 0.0625 = 2.625$. Also, the first decision possibility includes only one outcome where all vehicles are lost while the overall expected vehicle loss is $0.1875 + 0.125 + 0.1875 \times 2 + 0.0625 \times 2 + 0.0625 \times 3 = 0.9375$. Moreover, the outcomes where all demands have been served have a corresponding combined reward probability of 0.6875 and an expected cost of $56 \times 0.1875 + 58 \times 0.125 + 21 \times 0.375 = 25.625$. In contrast, for the second decision possibility, the outcomes provide a comparatively higher expected reward given by: $3 \times 0.89 + 2 \times 0.11 = 2.89$ (more demands can potentially be served). In addition, the second decision possibility includes two outcomes where all vehicles are lost while the overall expected vehicle loss is $0.05 \times 2 + 0.06 \times 2 + 0.18 + 0.15 + 0.05 \times 3 + 0.06 \times 3 = 0.88$. Moreover, the outcomes where all demands have been served, have a corresponding higher combined reward probability of 0.89 albeit having a comparatively higher expected cost of $93 \times 0.05 + 98 \times 0.06 + 61 \times 0.18 + 56 \times 0.15 + 26 \times 0.45 = 41.61$.

The first decision possibility, has two outcomes with the smallest probability of 0.0625 and corresponding reward/cost ratios of $\frac{1}{113}$ and respectively $\frac{2}{86}$ while having only one outcome with the highest probability of 0.375 and a reward/cost ratio of $\frac{3}{21}$. For the second decision possibility, there are also two outcomes with the smallest probability of 0.05 and a corresponding reward/costs ratios of $\frac{2}{120}$ and respectively $\frac{3}{93}$ while having only one outcome with the highest probability of 0.45 and a reward/cost ratio of $\frac{3}{26}$. We can note that the smallest as well as the highest probability outcomes belong to the second decision possibility. We can observe that there are specific trade-offs from the perspective of a decision maker that seeks to have all demands served while keeping the delivery cost as low as possible and avoiding the penalties resulting from lost vehicles. The first decision possibility exhibits a mix of outcomes with comparatively lower expected cost, lower potential vehicle losses but also a lower combined probability of serving all demands. The second decision making possibility also exhibits a mix of outcomes with comparatively higher expected cost, higher potential vehicle losses but also a higher combined probability of serving all demands. All this hints at the need to employ an elaborated analysis for decision making.

Thus, when considering only an EMC-based evaluation, given that $EMC_1 < EMC_2$, a decision maker would favour the first decision possibility with the lower EMC value of 53.625.

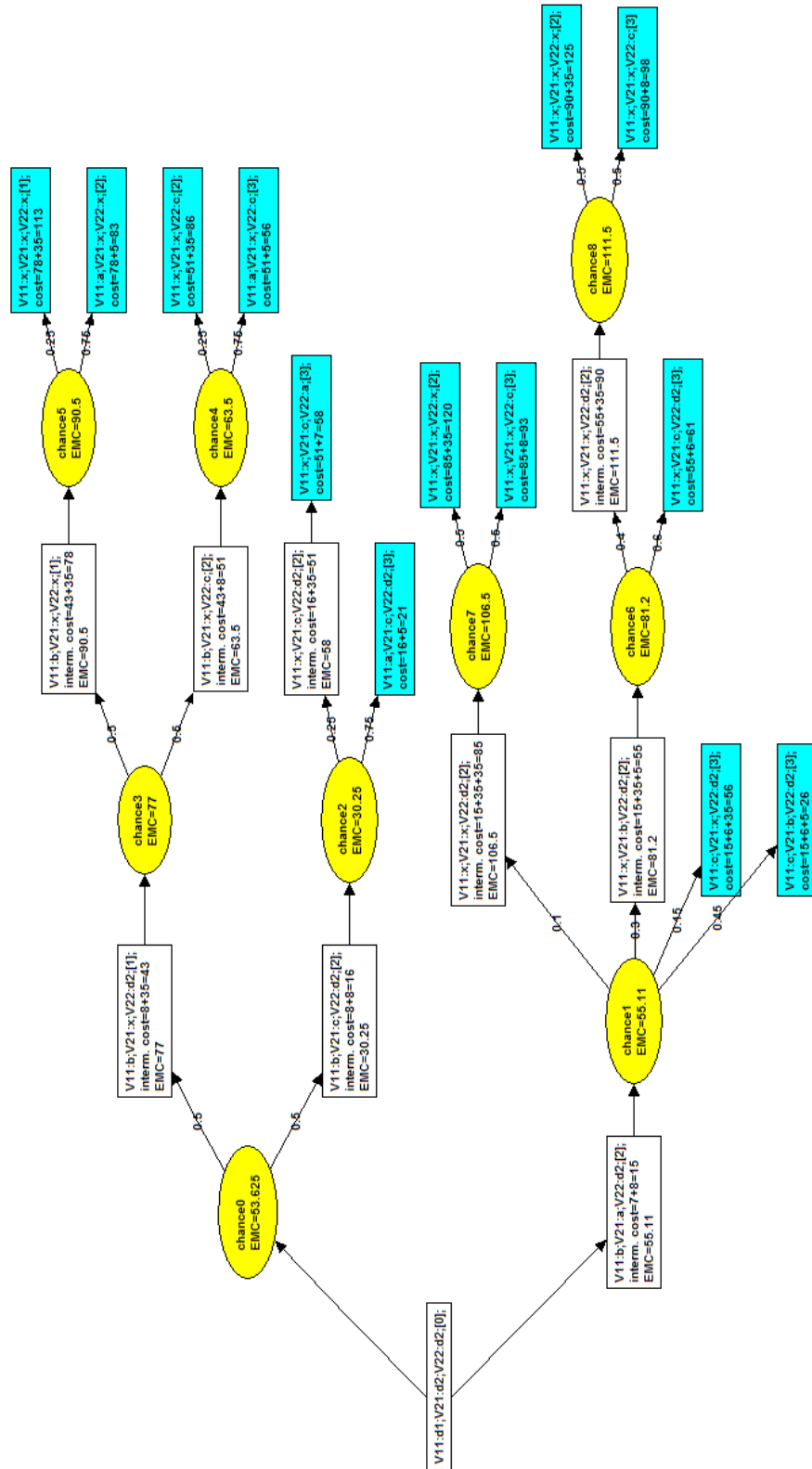


Figure 3.4: Decision tree with EMC

However an EMC-only analysis whereby a lower estimated cost is favoured, may not be always adequate. Consequently, a more elaborated analysis involving a cost threshold (maximum allowed budget) and a cost utility may provide better decision making support. The updated decision tree with cost threshold of 90 and cost utility annotation is depicted in Figure 3.5 where the yellow ovals represent chance nodes, the red rectangles represent unfavourable outcomes, the amber rectangles stand for fair outcomes and the green rectangle stands for the most favourable outcome (having the highest utility). The aforementioned decision tree is obtained from the initial decision tree annotated with EMC, for which all the branches where the cost is exceeding a value of 90 are removed. This limits potential excessive losses from certain decision making considerations that may lead to increasing the cost over the allowed budget (e.g., some contingency options may not be exercised). Each decision tree outcome also lists its probability of occurrence corresponding to its respective chance node chain. Then, we assign the utility cost uc as the ratio of the probability utility value u (corresponding to the likelihood of occurrence p for a given outcome as shown in Table 3.2) and the *cost* of reaching that outcome: $uc = u/cost$. In this setting, a decision maker favours a higher uc value.

For an outcome likelihood of less than 10%, we assign a probability utility value of 0, meaning that we find no value in any outcome with a probability of less than 10%. Conversely, for outcome likelihoods ranging from 10% to 100%, we assign various probability utility values ranging from 10 to 100, according to the breakdown provided in Table 3.2. In order to better convey the concept, the probability utility is normalized, ranging from 0 (no value) to 100 (maximum value). We also note that the values in Table 3.2 reflect the consideration of a certain degree of risk aversion from the decision maker side whereby low probability intervals are assigned comparatively lower utility values (e.g., 0 utility value for probabilities in the interval $0 \leq p < 10\%$; 10 utility value for probabilities in the interval $10\% \leq p < 15\%$). We propagate the corresponding euc (estimated utility cost) values obtained up toward the root of the decision tree in a similar manner to the EMC values. By observing the updated decision tree with estimated utility cost annotation, we can notice that its second decision possibility has a better (higher in this case) estimated utility:

$$euc_1 = 0.262 \times 0.5 + 1.828 \times 0.5 = 1.045$$

compared to the second one:

$$euc_2 = 0.117 \times 0.1 + 0.240 \times 0.3 + 0.357 \times 0.15 + 2.307 \times 0.45 = 1.175.$$

Outcome Probability Range	Outcome Probability Utility (u)
$0 \leq p < 0.10$	0
$0.10 \leq p < 0.15$	10
$0.15 \leq p < 0.30$	20
$0.30 \leq p < 0.45$	50
$0.45 \leq p < 0.60$	60
$0.60 \leq p < 0.75$	70
$0.75 \leq p < 0.90$	80
$0.90 \leq p < 0.99$	90
$0.99 \leq p \leq 1$	100

Table 3.2: Probability utility

The rewards corresponding to the possible outcomes of the updated first decision are as follows:

- 3, with a combined probability given by:

$$\underbrace{(0.5 \times 0.5 \times 0.75)}_{0.1875 \text{ (penalty 35)}} + \underbrace{(0.5 \times 0.25)}_{0.125 \text{ (penalty 35)}} + \underbrace{(0.5 \times 0.75)}_{0.375 \text{ (penalty 0)}} = 0.6875;$$

- 2, with a combined probability given by: $\underbrace{(0.5 \times 0.5 \times 0.25)}_{\text{(penalty } 2 \times 35)} = 0.0625$ and
- 1, with a probability given by: $\underbrace{(0.5 \times 0.5)}_{\text{(penalty } 2 \times 35)} = 0.25$.

Moreover, the rewards corresponding to the possible outcomes of the updated second decision are:

- 3, with a combined probability given by:

$$\underbrace{(0.3 \times 0.6)}_{0.18 \text{ (penalty 35)}} + \underbrace{0.15}_{\text{(penalty 35)}} + \underbrace{0.45}_{\text{(penalty 0)}} = 0.78 \text{ and}$$

- 2, with a combined probability given by: $\underbrace{0.1}_{\text{penalty } 2 \times 35} + \underbrace{(0.3 \times 0.4)}_{0.12 \text{ (penalty } 2 \times 35)}} = 0.22$.

The updated first decision possibility, has one outcome with the smallest probability of 0.0625 and corresponding reward/cost ratio of $\frac{2}{86}$ and one outcome with the highest probability of 0.375 and a reward/cost ratio of $\frac{3}{21}$. The updated second decision possibility, also has one outcome with the smallest probability of 0.1 and a corresponding reward/costs ratio of $\frac{2}{85}$ along with one outcome with the highest probability of 0.45 and a reward/cost ratio of $\frac{3}{26}$. The smallest probability outcome of 0.0625 belongs to the updated first decision and has a higher cost value ($cost = 86$) due to the loss of two vehicles in this outcome. The small probability value of the least likely outcome corresponds to having 0 utility ($u = 0$) and 0 utility cost ($uc = 0$). In contrast,

the highest probability outcome of 0.45 belongs to the updated second decision and has a smaller cost value ($cost = 26$) since no vehicle is lost in this outcome. The probability value of the most likely outcome corresponds to having most utility ($u = 60$) and a related $uc = 2.307$ utility cost.

For the updated first decision possibility, the outcomes provide an overall expected reward given by: $3 \times 0.6875 + 2 \times 0.0625 + 1 \times 0.25 = 2.4375$. Also, the updated first decision possibility includes no outcome where all vehicles are lost while the overall expected vehicle loss is $0.1875 + 0.125 + 0.0625 \times 2 + 0.25 \times 2 = 0.9375$. The outcomes where all demands have been served maintain a corresponding combined reward probability of 0.6875 as well as an expected cost of $56 \times 0.1875 + 58 \times 0.125 + 21 \times 0.375 = 25.625$. In contrast, for the updated second decision possibility, the outcomes provide a comparatively higher expected reward given by: $3 \times 0.78 + 2 \times 0.22 = 2.78$ (more demands can potentially be served). The updated second decision possibility includes no outcomes where all vehicles are lost while the comparatively lower overall expected vehicle loss is $0.18 + 0.15 + 0.1 \times 2 + 0.12 \times 2 = 0.77$. Furthermore, the outcomes where all demands have been served, have a somewhat higher combined reward probability of 0.78 coupled with a slightly higher expected cost of $61 \times 0.18 + 56 \times 0.15 + 26 \times 0.45 = 31.08$.

The foregoing results indicate that from the perspective of considering an allowable budget along with an utility function, a decision maker would favour the second decision making possibility ($euc_2 > euc_1$) with a corresponding estimated utility cost of 1.175.

Thus, a noteworthy remark is that a more calculated decision support can be provided by factoring the cost utility when compared to using only the EMC value. The initial assessment that the first decision would be favourable is overturned by the more elaborated use of the cost utility whereby the second decision is the recommended one.

3.7 Property Specification

For property specification, we mainly make use of the until operator \mathcal{U} in order to evaluate the probability of reaching different situations of interest. Figure 3.6 shows the module editor of PRISM while Figure 3.7 depicts the property specification panel of PRISM.

Figure 3.8(a) depicts the evaluation of the minimum and the maximum probabilities of serving all demand nodes.

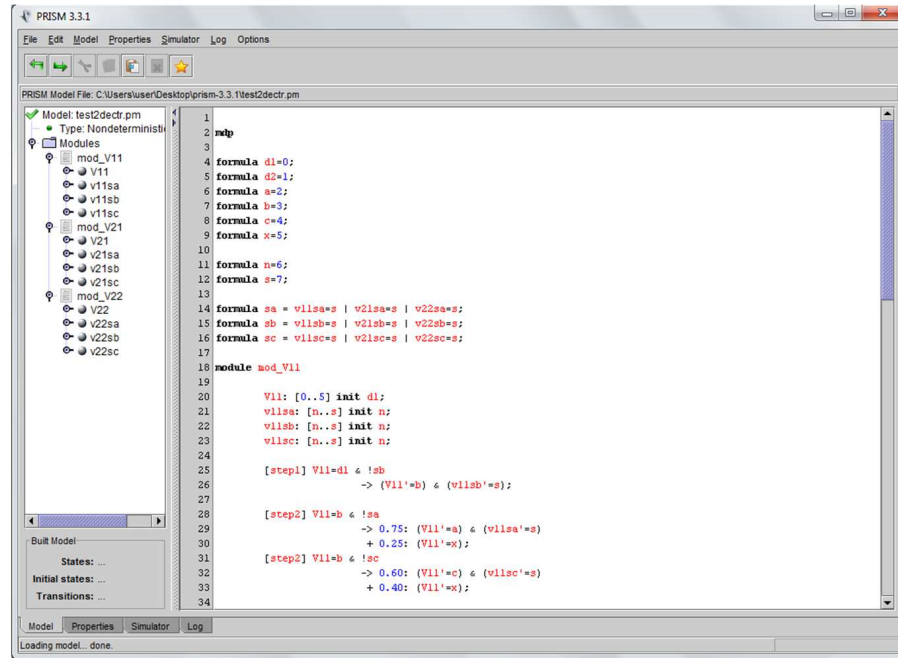


Figure 3.6: PRISM model-checker environment - Module editor

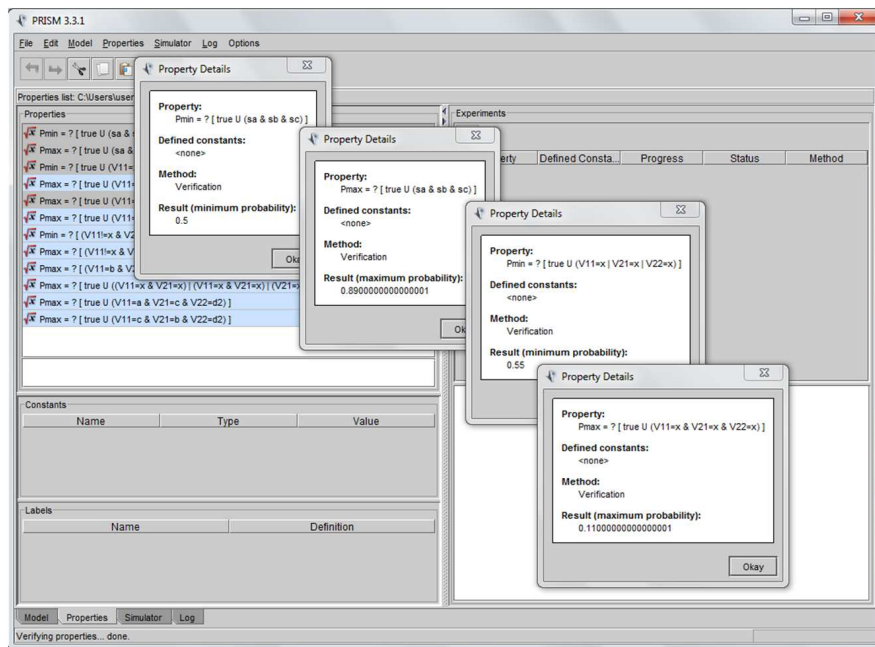


Figure 3.7: PRISM model-checker environment - Property specification and verification

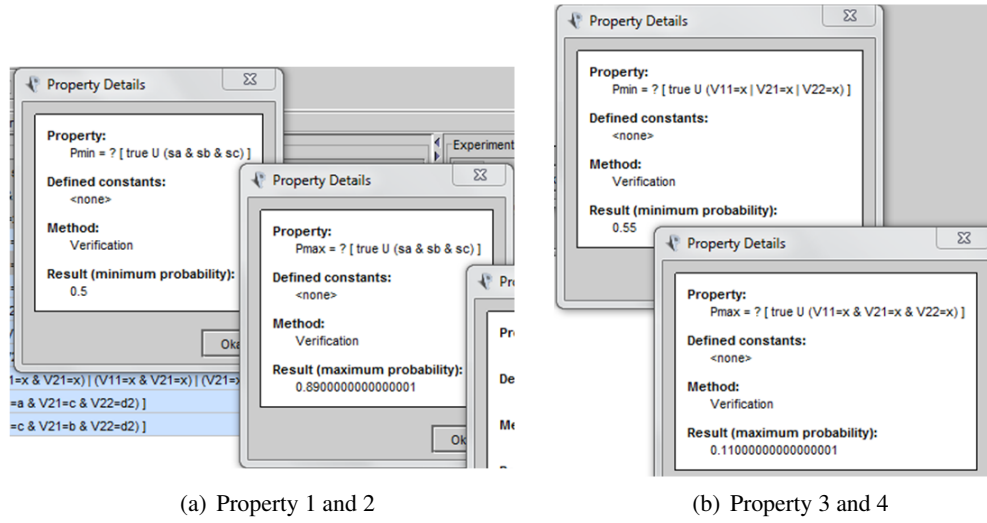


Figure 3.8: PRISM model-checker environment - Property verification results

The corresponding properties are:

Property 1: $P_{min} = ? [\text{true } U (sa \ \& \ sb \ \& \ sc)]$

Property 2: $P_{max} = ? [\text{true } U (sa \ \& \ sb \ \& \ sc)]$

As we can see in Figure 3.8(a), the minimum probability value evaluates to 0.5, meaning that there is at least 50% chance to serve all demand nodes as required by the system assumptions. The maximum probability value evaluates to 0.89, meaning that there is at most 89% chance of serving all demand nodes. Furthermore, the evaluation of the minimum probability of losing at least one vehicle and the maximum probability of losing all vehicles is depicted in Figure 3.8(b) using the properties:

Property 3: $P_{min} = ? [\text{true } U (V11=x \ | \ V21=x \ | \ V22=x)]$

Property 4: $P_{max} = ? [\text{true } U (V11=x \ \& \ V21=x \ \& \ V22=x)]$

As we can see in Figure 3.8(b), the minimum probability value evaluates to 0.55, meaning that there is a 55% chance to lose at least one vehicle. The maximum probability value of losing all vehicles evaluates to 0.11, meaning that there is at most 11% chance of losing all vehicles, which satisfies the requirements stated in the system assumptions. In order to show how different risk mitigation measures (e.g., extra vehicle availability) can be evaluated using the proposed

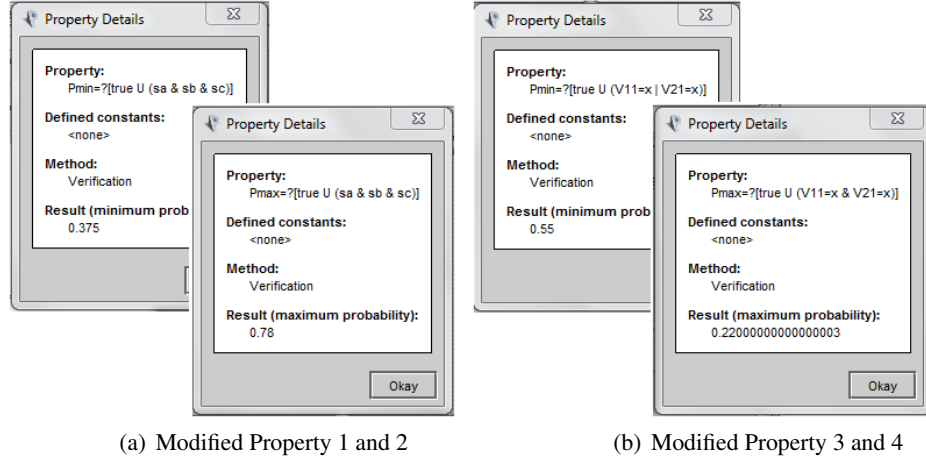


Figure 3.9: PRISM model-checker environment - Verification of modified properties

technique, we re-evaluate the same properties for a modified system where each depot has only one vehicle. The evaluation results are depicted in Figure 3.9(a) and respectively Figure 3.9(b). In the case of the latter, the property expressions are slightly changed in order to reflect the missing extra vehicle of depot 2:

Modified Property 3: $P_{min} = ? [\text{true } U (V11=x \mid V21=x)]$

Modified Property 4: $P_{max} = ? [\text{true } U (V11=x \& V21=x)]$

The property assessment of the modified system allows to appraise the degree to which the risk can be reduced by considering the availability of an additional vehicle in the second depot compared to the case where the additional vehicle is not available. For the modified model with only one vehicle in each of the depots, the minimum probability of serving all demand nodes is 0.375, meaning that there is at least 37.5% chance of serving all nodes. This value will not meet the requirements stated in the system assumptions. Also, the maximum probability of serving all demand nodes is 0.78, meaning that there is at most 78% chance of serving all demand nodes. We can notice that both the minimum and the maximum probability values are lower when compared to the case where the extra vehicle is available. The probability of losing at least one vehicle for the modified model is the same (55%) as in the case where there are two vehicles at depot 2 since in that case the extra vehicle will only be used as contingency in case that one or both of the other vehicles will be lost. However, the probability of losing all (both vehicles) is higher (22%) in this

case, which also does not satisfy the requirements of the system assumptions. The value is double compared to the case where risk reduction measures are in place (one extra vehicle at depot 2).

As shown in the foregoing, the proposed approach can be used to assess the satisfaction of system requirements such as acceptable risk exposure. Moreover, it provides the means for risk reduction appraisal following various risk mitigation measures such as the availability of supplementary vehicles, which represents a contingency feature that has been considered in the initial model.

3.8 Summary

This chapter presented an approach that leverages probabilistic model-checking in order to assess risk related properties for transportation tasks in the presence of a choice policy over different available routing options and various degrees of uncertainty. The proposed approach allows to evaluate risk exposure and contingency options for mitigating the risk of transportation tasks. Since the proposed approach employs probabilistic modelling it allows the assessment of properties formally expressed in probabilistic temporal logic. Transport tasks are captured using Markov Decision Process (MDP), which represents a behavioural model that can be subjected to probabilistic model-checking. This allows for formal verification of properties expressed in probabilistic computation tree logic (PCTL) capturing risk related outcomes along with system requirements of interest. In this setting, assessing properties related to possible outcomes can be very useful for tuning plan parameters and for contingency options evaluation. Relevant properties have been discussed for evaluating the probability of successful completion of transportation tasks along with lower and upper risk related cost bounds. Moreover, the usefulness of constructing decision trees based on model-checking results was shown to provide insightful means for risk appraisal. The enhancement of such decision trees was also discussed by factoring an utility function that can be used to reflect the degree of risk aversion that decision makers might have.

Compared to [88], which uses fuzzy logic for transportation risk assessment, the approach in this chapter leverages probabilistic modelling, thus allowing the assessment of properties formally expressed in probabilistic temporal logic. With respect to [116], which proposes a supply chain risk assessment method factoring the valuable albeit subjective experience of professional subject

matter experts, the approach in this chapter rests on formal verification which can provide more objective assessment. Moreover, in contrast to [32], which uses analytic network process for supply chain risk assessment by employing weights assigned by decision makers, the approach in this chapter is not subject to weight scale sensitivity.

The limitations of the proposed approach stem from the known limitations of model-checking in general, which relate to the problem of state explosion. Thus, for behavioural models with very large state spaces, techniques based on model-checking have associated scalability issues. In this respect, composing the dynamics of many vehicles with a very large number of route alternatives would be computationally prohibitive. Consequently, for large transportation plans the proposed approach can be most useful when applied over a selected set of route alternatives with associated risk factors. This can appropriately restrict vehicle dynamics and allow to explore a large number of meaningful route combinations while limiting state explosion.

The advantages of the approach in the context of transport task planning, consist in providing decision makers key capabilities for assessing risk related properties via probabilistic model-checking, which represents a proven technology for formal assessment of behavioural models. In addition, the ability to construct decision trees from model-checking results provides the practical advantage of facilitating decision making by offering an enhanced and more informative risk picture contrasting different outcomes of interest.

Chapter 4

Hierarchy-Aware Distributed Plan Execution Monitoring

The need for collaborative plan execution typically involving the provision of logistic support is of high interest in the context of operations management ranging from unit management to provisioning critical supplies as part of disaster relief efforts. An important aspect in this setting, is the aggregation of distributed nodes into clusters. This allows to localize the information exchange at the corresponding level of the distributed nodes. Moreover, cluster heads [74] can be used to aggregate the information relevant for the area covered by their respective clusters. Cluster heads can then further exchange information. Figure 4.1 depicts typical arrangements of distributed nodes that are aggregated into clusters and related cluster heads.

In this setting, this chapter is addressing the problem of shared information awareness in the context of distributed monitoring involving a hierarchy of distributed participants collaborating in the execution of a plan taking place in a hostile environment, prone to communication disruption.

We consider that multiple distributed participants (e.g., plan executing agents) are responsible to carry out the actions required for the plan. Moreover, the participants can support monitoring activities by sharing information (notify/report noteworthy events) to their cluster peers in pursuit of coordination while executing plan related actions. However, the participants must also respect a command hierarchy. In this respect, we bring forth the concept of asymmetric clustering, which involves non-reciprocal neighbourhood relationships among cluster peers and higher in command

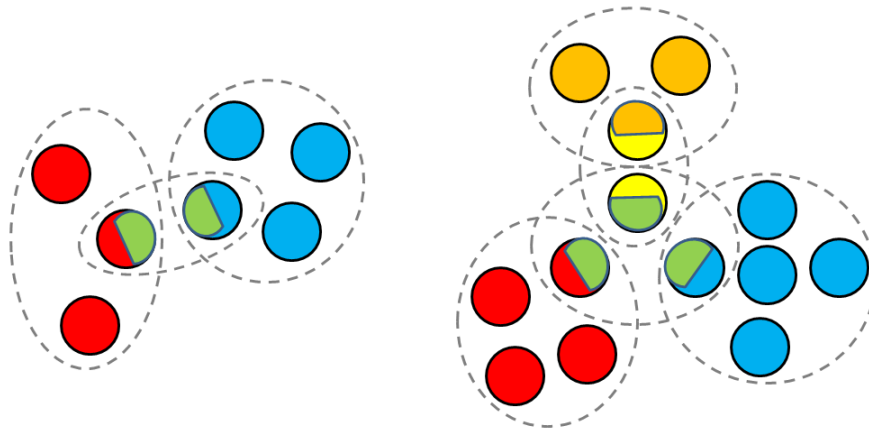


Figure 4.1: Example of hierarchical clustering for collaborative plan execution

participants. Asymmetric clustering allows for data exchange among coordinating peers on the same hierarchical level while propagating the information on the higher levels in command hierarchy. During execution, the environment might prompt the agents to deviate from their established actions thus potentially requiring plan adaptation and replanning. Thus, monitoring is necessary to correct the execution if needed. The aim is toward a lightweight, hierarchy-aware distributed monitoring protocol whereby the participants can use simple rules and procedures for effective communication in a hostile environment in order to obtain a high level information awareness of the overall situation. Moreover, in such hierarchy-aware distributed setting, the participants can join or leave a cluster at their appropriate hierarchical level thus mitigating the potential issue of single point of failure. For instance, the plan execution may lead an agent at tactical level to separate (break away) from a cluster, no longer needing to coordinate with that cluster, subsequently joining another cluster with which it may need to coordinate with.

The focus is on the following aspects of interest:

- Highlight the importance of collaborative plan execution in dynamic environments prone to communication disruption.
- Elaborate a hierarchy-aware distributed monitoring approach suitable for shared information awareness.

- Formalize the information sharing mechanism using communicating Markov Decision Processes in order to analyse the underlying dynamics using probabilistic model-checking.
- Assess the degree of information awareness via formal specifications expressed in probabilistic computation tree logic (PCTL).
- Conduct an illustrative case study in order to provide important observations with respect to the hierarchical aggregation of information and leveraging model-checking results to maximize an information awareness utility function.

4.1 Assumptions

Each participant is assumed to perform a specific plan related task in an uncertain environment. During task execution, each participant generates events resulting from deviations prompted by the environment (e.g., positional change to avoid obstacles, detected changes in the environment, etc). Thus, the events are assumed to relate to plan deviations (e.g., positional changes relative to established routing path in the case of vehicle routing) or changes in the environment (e.g., availability or unavailability of different route segments for logistic delivery). The event dissemination protocol assumes an initiating phase to establish communication setup. We also assume that each transmitted message includes a checksum allowing recipients to validate it.

4.2 Approach

We consider a decentralized setting whereby distributed participants aggregate into clusters based on neighbouring distance criterion, exhibiting asymmetric clustering to reflect the command hierarchy. In this asymmetric setup, lower (e.g., tactical) level peers can be neighbours to one or more higher (e.g., operational) level peers but the higher level peers are not neighbours to the lower level peers. All parties perform the same procedures to cooperatively maintain within their clusters a high level of information sharing. Given the plan execution environment nature in our scope of interest, techniques such as clustering and gossiping can support effective information sharing. These techniques can be useful to disseminate updates in a manner that mitigates the effects of the

erroneous communication and uncertainty in the environment. Figure 4.2 (a) depicts the participants' setup. In this context, gossiping is useful since it employs periodic communication of short peer-to-peer messages. This allows to mitigate message loss by periodic retransmission. Even though the overhead resulting from retransmission might represent a concern, it can be addressed by employing a validity period (fresh window) with respect to the events to be communicated. If an event exits the fresh window, it is considered as representing old information that is no longer relevant to be communicated. A more elaborated scheme may prioritize the dissemination of event information related to deviated actions that are prerequisite for subsequent actions. Also, to address situations where old (delayed) information could be of more importance because it is awaited by other participants, a validity interval (e.g., $[t_{current}, t_{current} + \Delta]$) may be added to each event to be disseminated (the disseminated information will be valid for Δ units of time from present time). Using such intervals, one can address both the obsolescence of information along with the priority of events to be disseminated. However, to present the approach succinctly we consider the less elaborated priority scheme. Figure 4.2 (b) depicts the fresh window usage. Each communication initiating participant can send information by pseudo-randomly selecting among other participants within an established clustering distance.

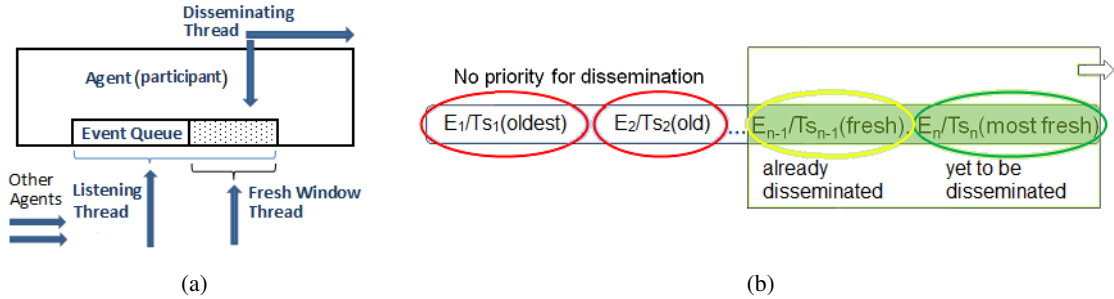


Figure 4.2: Participant threads/interactions (a) and event dissemination fresh window (b)

The generated events are placed in an event queue over which a fresh window slides. Each participant has three executing threads respectively for sending (disseminating), listening (receiving) and fresh window update. In Figure 4.2 (a), we observe that the fresh window thread shifts the fresh information window such that the events exiting the fresh window are essentially discarded as shown in 4.2 (b). Figure 4.2 (a) also shows that each participant has a listening thread whereby

it receives events from other participants. Each received event is placed in the event queue. Also, each participant runs a disseminating thread that inspects periodically (i.e., each time unit) the event queue in order to select an event to disseminate.

The aforementioned dissemination technique does not require a transmission acknowledgement mechanism and the erroneous communication is essentially discarded (e.g., in case of message checksum mismatch). The fresh event information is quickly disseminated while retransmission takes place periodically within each cluster.

4.3 Asymmetric Clustering for Hierarchical Command Chain

Asymmetric clustering essentially implies non-reciprocal neighbouring relations. In this respect, the participants are not having a homogeneous type but their assigned types reflect an established hierarchy (e.g., tactical, operational and strategic levels of command). In this setting, for an established clustering distance (e.g., distance to horizon for ground operations), the participants at the same level exhibit mutual neighbouring relations such that if A is neighbour to B, then B is also neighbour to A. Moreover, as can be seen in Figure 4.3 (a), depending on the clustering distance employed, A and B can be mutual neighbours, B and C can also be mutual neighbours but A and C may not be neighbours (A and C may be too far apart from each-other but each within clustering distance to B).

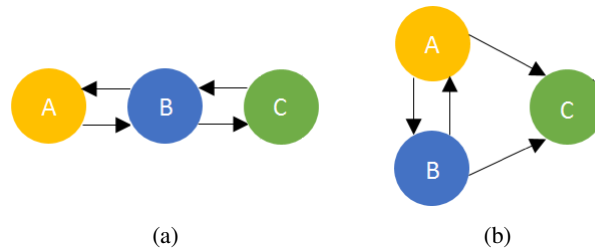


Figure 4.3: Symmetric membership across clusters (a) and symmetric/asymmetric membership in the same cluster (b)

The command hierarchy is modelled via asymmetric clustering where a group of peers at a lower level (e.g., tactical) are neighbours to one or more peers at a higher level (e.g., operational) but the higher level peers are not neighbours to the lower level peers (see Figure 4.3 (b)). This way,

lower level peers disseminate information to each other and also to the higher level participants that they are neighbours with. Conversely, the higher level participants receive but do not disseminate information to lower levels. This does not preclude the participants at higher levels to initiate communication with the lower levels in case that a change of plan is needed. It only reflects the flow of monitoring information respecting an established command hierarchy. Furthermore, while a subordinate typically reports and receives commands from only one superior, in distributed collaborative setting, it may also disseminate reports to other peers of its commanding superior in order to support replanning alternatives where a lower level peer loses contact with its original commanding superior. Thus, from monitoring standpoint, the event information is aggregated hierarchically such that the overall situational awareness is reflecting the appropriate decision making activity in the scope of the participants. For instance, two separate clusters that have a common higher level peer neighbour will not directly share information but the higher peer level will have a shared information awareness with both of these clusters. In this setup, we consider the information flow for the monitoring aspect in relation to achieving shared information awareness.

4.4 Protocol Formalization

Hierarchy-aware distributed gossip data exchange is captured with the following constructs:

- $N = \{n_k \mid k = 1 \dots m\}$, representing a set of participating nodes that can send and receive data (e.g., numeric, boolean, etc.);
- A distance relation over $N \times N$ that can be used to establish node clusters, with the ability of any node to be part of more than one cluster;
- A set of clusters $C = \{c_p \mid 1 < p < m\}$ with the property that $\forall p, c_p \subseteq N, \bigcup_p c_p = N$ and $\forall n_i, n_j \in c_p, |c_p| \geq 2, \text{dist}(n_i, n_j) \leq \text{clDist}$ where clDist represents an established clustering distance. We note that there is no hierarchy for the particular case where all nodes are in the same cluster with symmetric membership.

For any given cluster c_p , there is a partition of its elements into two subsets $c_p^s \subseteq c_p$ and $c_p^a \subseteq c_p$ such that $c_p = c_p^a \cup c_p^s$ and $c_p^a \cap c_p^s = \emptyset$. The two subsets stand for the symmetric and respectively asymmetric members of c_p .

For convenience of notation, we hereon make use of the following substitutions: $\forall i \in \{1..m\}$ where $n_i \in c_p^s$ and $\forall j \in \{1..m\}$ where $n_j \in c_p^a$, we then substitute $n_i \rightarrow n_i^s$ and $n_j \rightarrow n_j^a$, to mark the relationship of a node n_i or a node n_j as respectively symmetric or asymmetric to cluster c_p .

In the foregoing context, with respect to the information flow, the nodes can have one of two relations to a given cluster:

- *symmetric membership*: both send and receive updates to/from the cluster;
- *asymmetric membership*: only receive updates from the cluster.

In this setting, symmetric (same rank) nodes send updates to all the cluster members (symmetric and asymmetric) where they have symmetric membership. Conversely, the asymmetric (higher rank) nodes only receive updates from the cluster where they have asymmetric membership. However, each node receives updates from all the symmetric nodes in the clusters where it has (symmetric or asymmetric) membership. We assume that for any cluster where a node is selected to be a member of, the decision of its membership type (symmetric vs. asymmetric) is based on the hierarchical level of the underlying participant such that a node joins a cluster with asymmetric membership if at least one of the other cluster members has a lower rank on the command hierarchy. Otherwise the node joins with symmetric membership. We describe next the clustering relationships and their effect on the information flow from the perspective of each node. Each participant $n_k, k = 1..m$ is characterized by the tuple $\langle A_k, A_k^{DS}, A_k^{CS} \rangle$ where $A_k = n_k$ and:

- A_k^{DS} is the set of peers whose updates should be received by A_k (directly - from other peer cluster members or indirectly - from peers not directly clustered);
- A_k^{CS} is a *key/value* data store with *key* $\in C$ and *value* denoting symmetric (*s*), respectively asymmetric (*a*) relationship to the corresponding *key* clusters where A_k has membership. Since the membership relationship is either of type *s* or type *a*, then we can also substitute $\{s \rightarrow \text{true}, a \rightarrow \text{false}\}$.

Example: Let us consider a simple case of 4 communicating participants (1,2,3,4), of which (1,2,3) are on the same hierarchical level and (4) is on the next higher level. The participants are arranged in 2 clusters and their relationships are described by:

$N = \{n_1, n_2, n_3, n_4\}$, $C = \{(n_1^s, n_2^s), (n_2^s, n_3^s, n_4^a)\}$ with

$$\begin{aligned}
&\langle A_1 = n_1, \underbrace{\{n_1, n_2, n_3\}}_{A_1^{DS}}, \underbrace{(n_1, n_2) \rightarrow s}_{A_1^{CS}} \rangle \\
&\langle A_2 = n_2, \underbrace{\{n_2, n_1, n_3\}}_{A_2^{DS}}, \underbrace{(n_1, n_2) \rightarrow s; (n_2, n_3, n_4) \rightarrow s}_{A_2^{CS}} \rangle \\
&\langle A_3 = n_3, \underbrace{\{n_3, n_1, n_2\}}_{A_3^{DS}}, \underbrace{(n_2, n_3, n_4) \rightarrow s}_{A_3^{CS}} \rangle \\
&\langle A_4 = n_4, \underbrace{\{n_4, n_1, n_2, n_3\}}_{A_4^{DS}}, \underbrace{(n_2, n_3, n_4) \rightarrow a}_{A_4^{CS}} \rangle
\end{aligned}$$

Algorithm 1 Assessment of the expected information awareness level

```

1: Input:  $N, C$ 
2: for all  $k \in \{1..|N|\}$  do
3:   initialize  $tup = \langle n_k, DS = \{n_k\}, CS = \{\} \rangle$ 
4:   for all  $nCl \in C$  where  $n_k \in nCl$  do
5:     if  $tup.n_k \in nCl^s$  then
6:        $tup.CS.addEntry(nCl, true)$ 
7:     else
8:        $tup.CS.addEntry(nCl, false)$ 
9:     end if
10:  end for
11:  for all  $nCl \in C$  where  $n_k \in nCl$  do
12:     $walkCl(tup, nCl, C);$ 
13:  end for
14: end for

```

Algorithm 2 walkCl - Recursive walk over the clustering structure

```

1: Input:  $tup, nCl, C$ 
2: for all  $ns \in nCl$  do
3:   if ( $tup.DS$  not contains key  $ns$ ) and ( $ns \in nCl^s$ ) then
4:      $tup.DS.add(ns)$ 
5:     for all  $nsCl \in C$  where  $ns \in nsCl$  do
6:       if  $tup.CS$  not contains key  $nsCl$  then
7:          $walkCl(tup, nsCl, C);$ 
8:       end if
9:     end for
10:  end if
11: end for

```

The previous structures allow to reason on the information flow. Thus, for n_1 , apart from its data, it should receive updates from n_2 (directly) and from n_3 (indirectly via n_2). Likewise,

apart from its data, n_2 should receive updates from n_1 (directly) and from n_3 (directly). Also, n_3 , apart from its data, should receive updates from n_1 (indirectly through n_2) and from n_2 (directly). Moreover, n_4 , apart from its data, should receive updates from n_1 (indirectly via n_2) and directly from both n_2 and n_3 . However, n_4 sends no updates due to the asymmetric membership in cluster (n_2^s, n_3^s, n_4^a) .

4.5 Information Awareness Degree

In order to evaluate the degree of information awareness, one needs to ascertain for each node $n_k \in N$, $k = 1..|N|$ what information it has versus what information it should have. In the foregoing example, n_1, n_2, n_3 share information while n_4 only receives information from n_1, n_2, n_3 . Consequently, given the formal description of the nodes relationships, a procedure can be elaborated to ascertain what each node should know. Algorithm 1 and Algorithm 2 allow to construct $\langle A_k, A_k^{DS}, A_k^{CS} \rangle$ for arbitrary clustering arrangements. By establishing what information each node should receive from other nodes, one also knows what information each node forwards to the clusters where its membership is symmetric. However, in disrupted communication environment, sending information may fail to reach recipients with a certain probability. Thus, for a given communication error rate, various cluster configurations would be more favourable to achieve full information awareness in the scope of each participant. The formalization allows the use of probabilistic model-checking to assess various cluster configurations relative to the flow of information.

From the probabilistic models supported by PRISM, we selected MDP since the proposed protocol exhibits both probabilistic and nondeterministic behaviours. The probabilistic behaviour mainly results from communication disruption, which is characterized by the error rate for receiving data. The nondeterministic behaviour stems from multiple communication choices available for a node that belongs to more than one cluster. With respect to the timing aspect of the protocol, we assume that each node synchronizes its time with a system timer (e.g., in practice one can employ clock synchronization via GPS or other time reference available) such that the time increments with the system time unit for each participant. Data communication among nodes occurs in out-of-order fashion and periodically such that each node sends data once per time unit.

The logic required for each participant node is described in a distinct module according to its

clustering relationships and we employ the capabilities of PRISM in order to compose the overall dynamics. Then, we can assess the protocol using probabilistic model-checking for various levels of disruption (increasing error rate values) by specifying corresponding properties capturing the realization of the expected information awareness at the top level in the hierarchy. Without loss of generality, communication disruption is modelled with a common error rate across all the modules in order to illustrate the approach more clearly. However, different error rates could be specified for each module if needed. Since we employ MDPs as semantic interpretation for the communication interaction, we express the required properties in PCTL. We use the bounded until operator \mathcal{U}' to evaluate the probabilities to reach different levels of information awareness for different fresh window sizes and disruption levels.

Example. We illustrate in Figure 4.4 the state diagram corresponding to the cluster arrangement in Figure 4.3 (b), which depicts one cluster (A^s, B^s, C^a) where the membership of A and B is symmetric while the membership of C is asymmetric. In this context, Figure 4.5 illustrates the state diagram for the system fresh window update.

Discussion. The previous example shows the key features of modelling communication via MDP:

- The model considers a snapshot of participants in a particular clustering arrangement and event information, which is abstracted for each participant as known/unknown information (e.g., Participant A initially knows a but not b, which is captured in the initial state of A as $A_va = \text{true}$ respectively $A_vb = \text{false}$).
- A system module (sys) is used for time reference synchronization among modules and fresh window update (up to a maximum of maxT units).
- Participants are represented by modules that interact with the system module for time synchronization and with other modules for data exchange (each symmetric participant sends data in its cluster once per time unit).
- Symmetric cluster membership results in modules A and B both sending and receiving data within cluster (A,B,C). However, data is lost with error rate err and successfully received with $\text{noerr}=1-\text{err}$ rate (for brevity, the state diagram omits self-transitions to the same state when erroneous communication occurs).

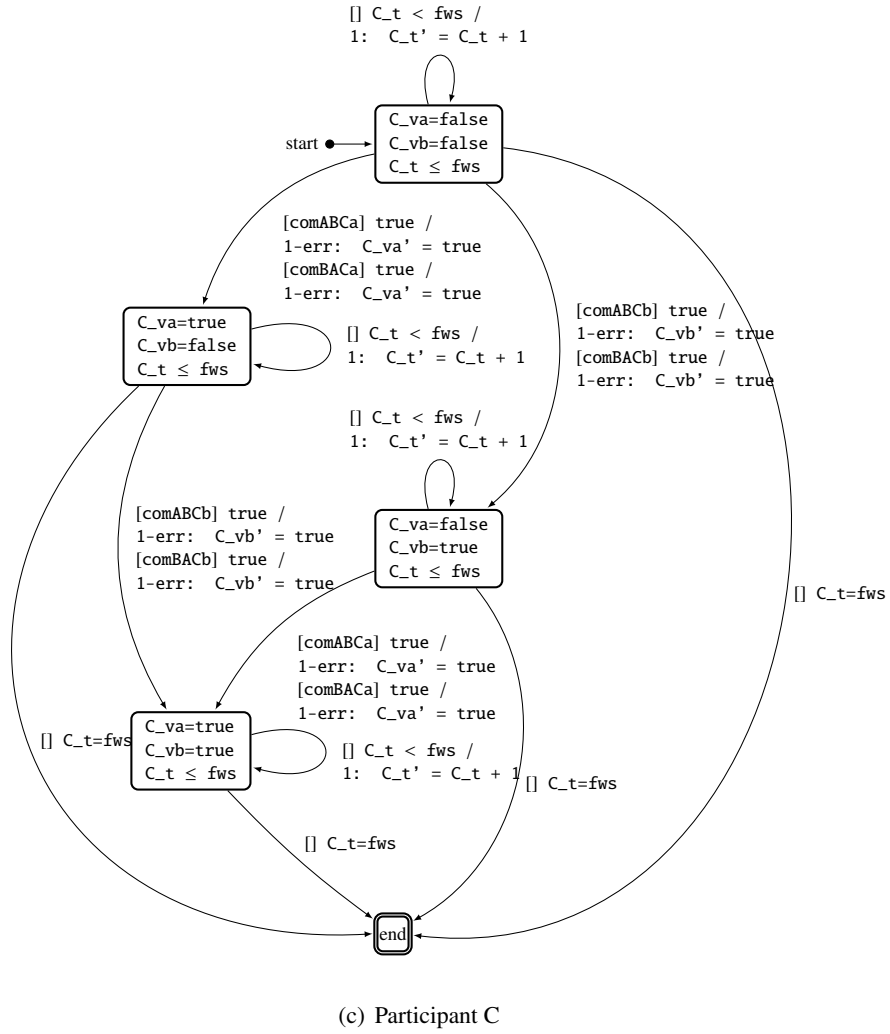
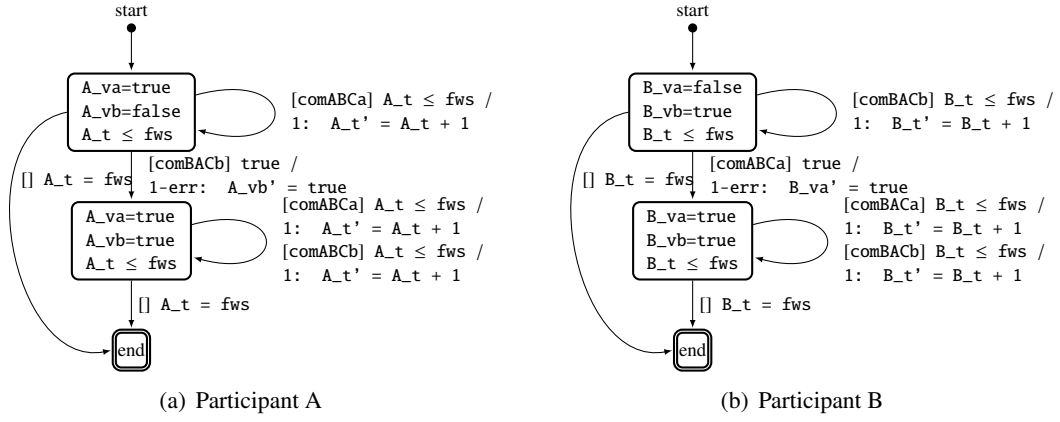


Figure 4.4: State diagrams corresponding to the participants in cluster (A^s, B^s, C^a)

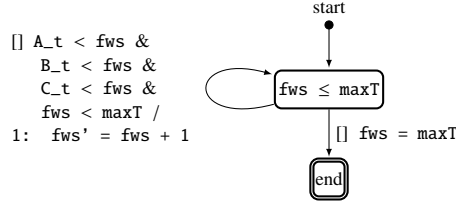


Figure 4.5: State diagram corresponding to the system fresh window size update

- Asymmetric cluster membership results in module C only receiving data within cluster (A,B,C). In this respect, the model captures the information received from A and B via C_va and respectively C_vb. Also, there is no need to capture Participant C own information since the latter is not disseminating its own data.

Probability assessment. In order to assess probabilities with PRISM, we specify corresponding PCTL properties capturing the realization of full information awareness. Thus, in the case of the previous example, we specify the PCTL property for assessing the probability of C eventually knowing the information of both A and B for various fresh window sizes and communication error rates: $P = ? [\text{true } \mathcal{U}^{t \leq fwr} (C_{va} \wedge C_{vb})]$, which in the PRISM environment turns into $P = ? [\text{true } \mathcal{U} \leq fwr (C_va \ \& \ C_vb)]$.

The aforementioned property can also be expressed using the finally (F) operator given the PCTL equivalence $(P = ? [\text{true } \mathcal{U} \ q]) \equiv (P = ? [\text{F } q])$. However, we use the until (U) operator since it is more general and required as part of PCTL syntax definition while (F) provides in essence syntactic sugar, which allows to improve the readability of complex properties.

The results of property verification provided by PRISM are presented in Figure 4.6, which shows the evolution of the probability for C achieving full information awareness in various disruption conditions (from 0% up to 90% error rate, in 10% increments) with increased fresh window size. An important observation is related to the nondeterminism present in the model. In general, MPDs have minimum and maximum probabilities for reaching a particular state depending on the available nondeterministic choices. However, in our case, we employ fairness since otherwise the model-checker may consider repetitive choices whereby a module preferentially synchronizes with some modules but not others in the case where multiple synchronizing choices are available. This hints to the need of employing fairness.

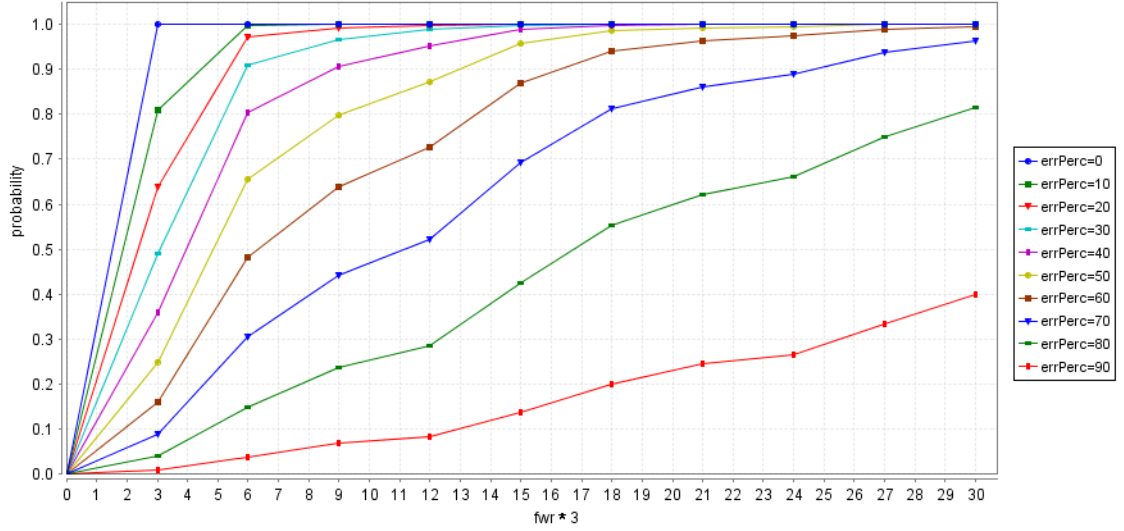


Figure 4.6: Probability assessment example

Another noteworthy observation is related to the fact that when assessing bounded time properties, the model-checker takes into account each transition in the model. However, since we synchronize the time of the participants (A_t , B_t , C_t) with the system reference (Sys), each module has to perform an individual transition in order to update the time in the whole model. Thus, for each time unit elapsed in the modules, the model-checker counts a number of model-checking time units equal to the number of participants. Therefore, in Figure 4.6 we note on the abscissa that the fresh window range (fwr) is multiplied by the number of participants (3). Consequently, the actual fresh window size should be divided accordingly by the number of participants. We can see that when there is no disruption (zero error percentage), full information awareness is obtained with 100% probability for very small fresh window size whereas for increasing levels of disruption, the probability decreases. Moreover, for a small cluster and a given disruption level, one can derive a mathematical relation between the window size and the probability of achieving information awareness. For instance, given a disruption level of 70% (as shown by corresponding graph in Figure 4.6 where $errPerc=70$), we can derive for cluster (A^s, B^s, C^a) a quadratic approximation for the probability of achieving information awareness as follows: $p = -0.0086fws^2 + 0.1861fws - 0.038$ with a coefficient of determination $R^2 > 0.99$. We note that this quadratic relation is only valid up to the maximum value of $30/3=10$ considered for

the fresh window size (fws). However, for more elaborated clustering arrangements, potentially involving dissimilar disruption levels in some of the clusters, no simple mathematical relation can be derived with high coefficient of determination as highlighted in the case study.

4.6 Case Study on Assessing Shared Information Awareness

We present an illustrative case study involving the clustering arrangement that is depicted in Figure 4.7, which shows a number of 6 participants (A, B, C, D, E, F) with the following hierarchy: A and B are on the same level and both are subordinated to C, which is on the same hierarchical level with D. The latter has two subordinates, namely E and F, each of which forms a separate cluster with D. We note that from the perspective of A and B, they are neighbours with C while

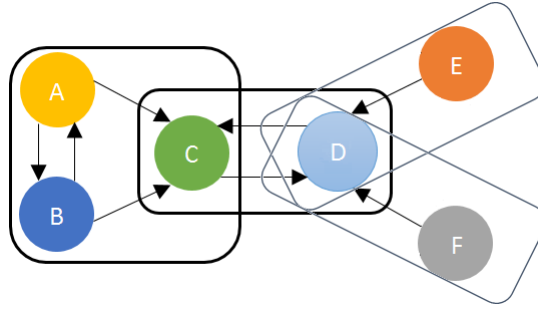


Figure 4.7: Asymmetric clustering for hierarchical information dissemination

from the perspective of C, the latter is not neighbour with A nor B (asymmetry). This means that A and B will disseminate information to each-other and to C but C will not share information with neither A nor B. Likewise, E and F are each neighbours with D while D is not a neighbour of E nor of F (asymmetry). This means that D will not share information with neither E nor F. From the perspective of C, it is a neighbour of D and similarly, from the perspective of D it is a neighbour of C. Thus, C and D are mutual neighbours. Hence, C and D will disseminate information only to each-other. The corresponding notation for the clustering arrangement is as follows:

$$N = \{A, B, C\}, C = \{(A^s, B^s, C^a), (C^s, D^s), (D^a, E^s), (D^a, F^s)\}.$$

After applying Algorithm 1, with input N and C , we obtain:

$$\begin{array}{ll}
\langle A_1 = A, \underbrace{\{A, B\}}_{A_1^{DS}}, \underbrace{(A, B, C) \rightarrow s}_{A_1^{CS}} \rangle & \\
\langle A_2 = B, \underbrace{\{B, A\}}_{A_2^{DS}}, \underbrace{(A, B, C) \rightarrow s}_{A_2^{CS}} \rangle & \\
\langle A_3 = C, \underbrace{\{C, A, B, D, E, F\}}_{A_3^{DS}}, \underbrace{(A, B, C) \rightarrow a; (C, D) \rightarrow s}_{A_3^{CS}} \rangle & \\
\langle A_4 = D, \underbrace{\{D, A, B, C, E, F\}}_{A_4^{DS}}, \underbrace{(C, D) \rightarrow s; (D, E) \rightarrow a; (D, F) \rightarrow a}_{A_4^{CS}} \rangle & \\
\langle A_5 = E, \underbrace{\{E\}}_{A_5^{DS}}, \underbrace{(D, E) \rightarrow s}_{A_5^{CS}} \rangle & \\
\langle A_6 = F, \underbrace{\{F\}}_{A_6^{DS}}, \underbrace{(D, F) \rightarrow s}_{A_6^{CS}} \rangle &
\end{array}$$

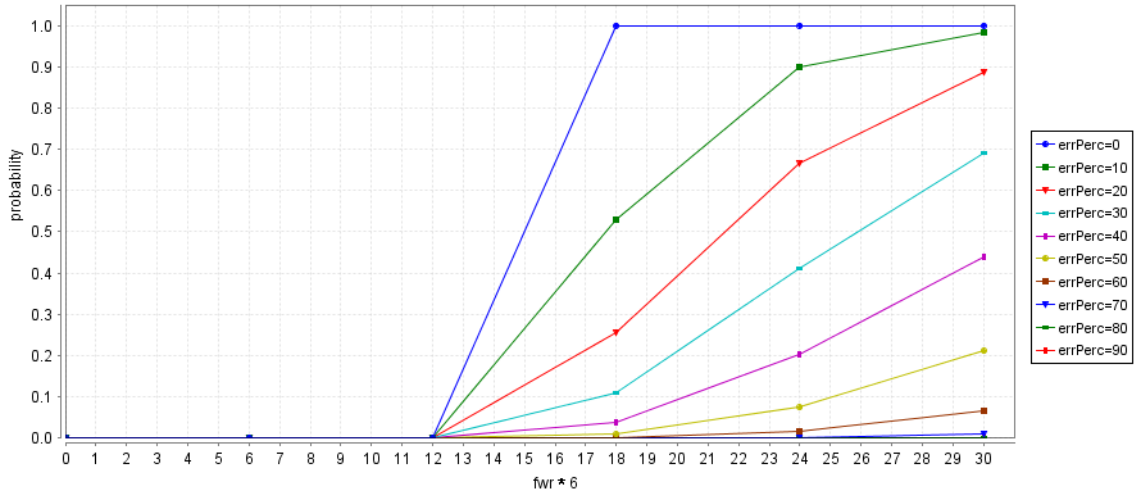


Figure 4.8: Probability assessment on achieving fully shared information awareness for participants C and D

We first specify the PCTL property for assessing the probability of both C and D eventually knowing the information of the other 5 participants: A, B, D, E, F and respectively A, B, C, E, F corresponding to fully shared information awareness or 5/5: $P = ? [\text{true } \mathcal{U}^{t \leq fwr} (C_{va} \wedge C_{vb} \wedge C_{vd} \wedge C_{ve} \wedge C_{vf} \wedge D_{va} \wedge D_{vb} \wedge D_{vc} \wedge D_{ve} \wedge D_{vf})]$, which in the PRISM environment is written as:

$P = ? [\text{true } \mathcal{U}^{t \leq fwr} (C_va \ \& \ C_vb \ \& \ C_vd \ \& \ C_ve \ \& \ C_vf \ \& \ D_va \ \& \ D_vb \ \& \ D_vc \ \& \ D_ve \ \& \ D_vf \)]$ (Property 1).

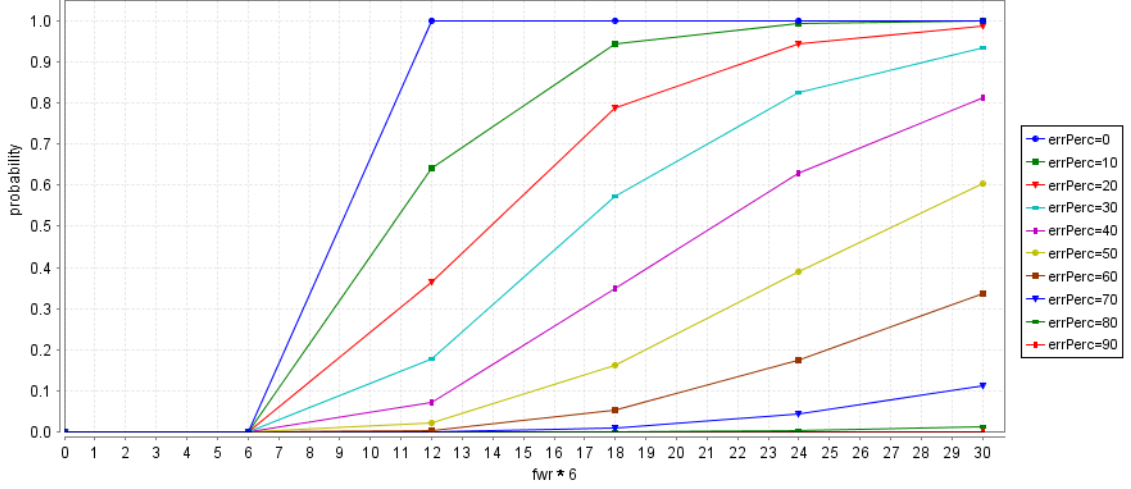


Figure 4.9: Probability assessment on achieving partially shared information awareness for participants C and D

Figure 4.8 shows the assessment result of Property 1 produced by PRISM for various fresh window sizes and error rates corresponding to increasing levels of disruption. The window size range axis is labelled by `fwr*6` since we have a total of 6 participants. We can see that the probability of achieving fully shared information awareness is notably affected in the presence of increased disruption. Since a partial shared information awareness might also be useful if it can be achieved more readily, we specify next the PCTL property for assessing the probability of C and D eventually knowing the information of 4 out of the other 5 participants (4/5). Given that C obtains the information of E and F through D and that D obtains the information of A and B through C, a partially shared information awareness of 4/5 corresponds to C eventually knowing the information of A, B, C, D, (E or F) and D eventually knowing the information of (A or B), C, D, E, F: $P = ? [true \mathcal{U}^{t \leq fwr} (C_{va} \wedge C_{vb} \wedge C_{vd} \wedge (C_{ve} \vee C_{vf}) \wedge (D_{va} \vee D_{vb}) \wedge D_{vc} \wedge D_{ve} \wedge D_{vf})]$, which in the PRISM environment becomes:

$$P=?[true \ U \leq fwr \ (C_va \ \& \ C_vb \ \& \ C_vd \ \& \ (C_ve \ | \ C_vf) \ \& \ (D_va \ | \ D_vb) \ \& \ D_vc \ \& \ D_ve \ \& \ D_vf \)] \text{ (Property 2).}$$

Figure 4.9 shows the assessment results for Property 2 corresponding to partially shared information as produced by PRISM for various fresh window sizes and error rates corresponding to

increased disruption levels. The window size range axis is likewise labelled by $fwr*6$ since there are 6 participants in total. We can see that the probability of achieving partial shared information awareness is less affected by the increased disruption when compared to the results for Property 1.

Figure 4.10 contrasts the probability profiles for achieving full and partial information awareness for a fresh window size of 5 ($fws*6=30$) and increasing error levels from 0% to 90% in 10% increments. The solid line depicts the probability profile for achieving full information awareness based on the data showed in Figure 4.8 for $fws*6=30$. The dotted line depicts the probability profile for achieving partial information awareness based on the corresponding data showed in Figure 4.9 for $fws*6=30$. The bar graph provides the difference between the values of the dotted graph and solid graph. We can note that the largest difference occurs at the error level of 50%, meaning that for a disruption level around 50%, partial information awareness can be achieved with the highest gain in probability compared to achieving full information awareness. In contrast, the difference is very small at the extremes, meaning that for small or high error levels, full information awareness can be obtained with a little lower probability when compared to achieving partial information awareness.

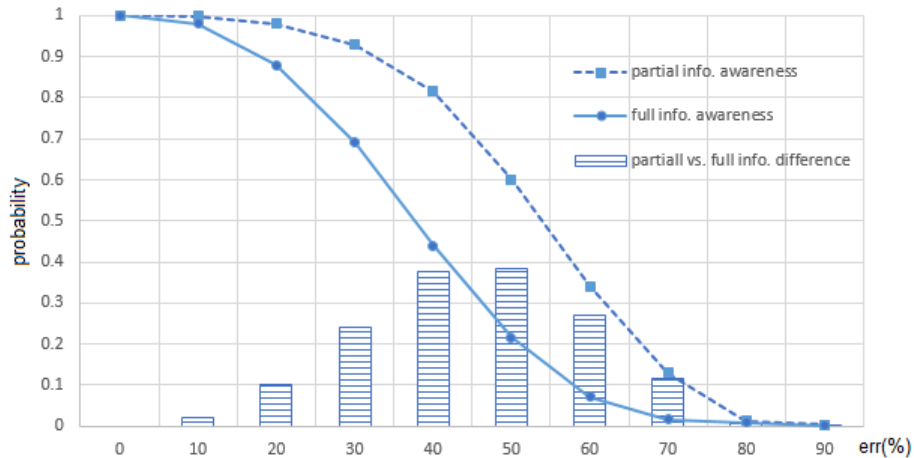


Figure 4.10: Probability to achieve full/partial information awareness for $fws = 5$ and increasing error rate levels

We show next two situations where some participants break from their clusters, leading to the formation new cluster arrangements as depicted in Figure 4.11 (up and down). Thus, in the upper side we have node A that changes position such that it no longer remains in clustering distance

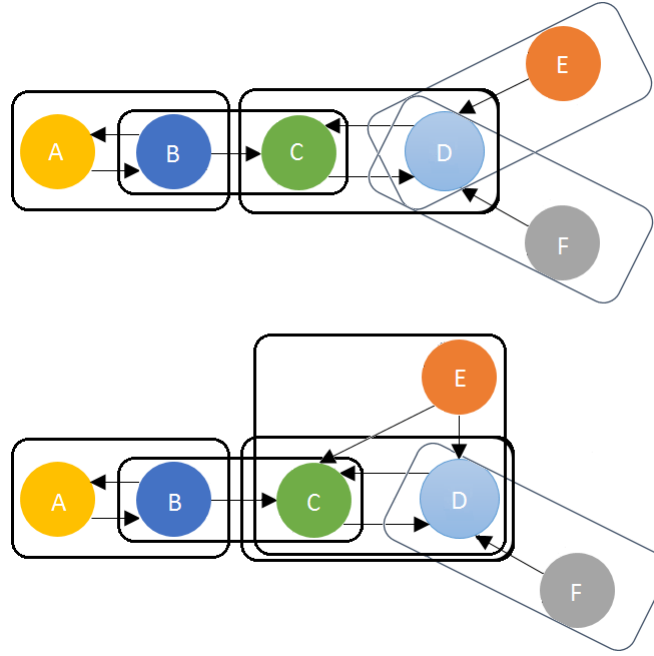


Figure 4.11: Participants changing position and forming new cluster arrangements

with C while still remaining in a cluster with B. Thus, the corresponding clustering arrangement for Figure 4.11 (up) is: (A,B),(B,C),(C,D), (D,E),(D,F). In the lower side of the figure, we can see a subsequent move of node E that changes position so that it arrives sufficiently close to both D and C to be part of a common cluster. Consequently, the corresponding clustering arrangement for Figure 4.11 (down) is: (A,B),(B,C),(CD),(C,D,E),(D,F). In the aforementioned arrangement, we note that cluster (C,D) is still present even though there is another cluster (C,D,E), which includes both C and D. This is due to the fact that in cluster (C,D,E), C and D can only join with asymmetric membership since E has lower rank. However, in order for C and D to exchange information they need to be part of a separate cluster (C,D) where they can have both symmetric membership.

We contrast in Figure 4.12 the degree of achieving full and partial shared information awareness in different cluster arrangements by comparing the initial clustering arrangement in Figure 4.7 against the clustering arrangements in Figure 4.11 (up and down) for a nominal disruption level of 50% error rate. The horizontal axis provides the the fresh window range (the window size values are multiplied by number of participants as previously discussed), while the vertical axis stands for the probability of achieving shared information awareness. We can see

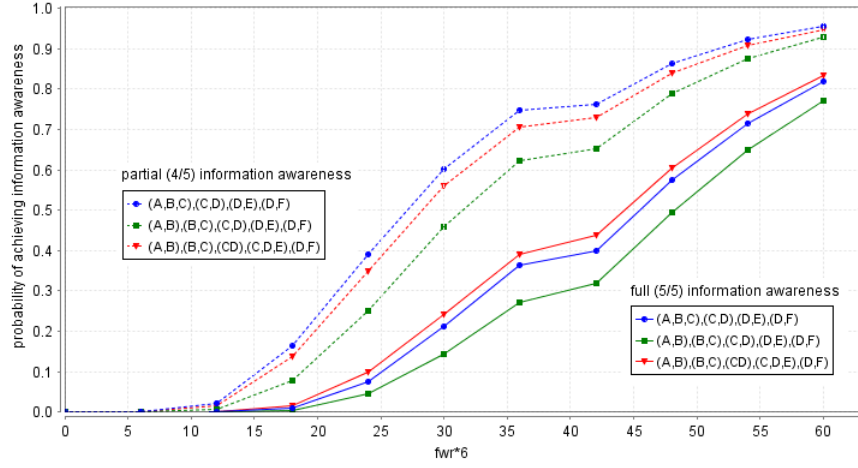


Figure 4.12: Comparison of shared information awareness in different cluster arrangements

that for a given fresh window size, the arrangement $(A,B),(B,C),(CD),(C,D,E),(D,F)$ has greater probability of achieving full information awareness (solid line), when compared to the initial $(A,B,C),(C,D),(D,E),(D,F)$ arrangement. Conversely, we note that for partial information awareness (dotted line), the arrangement $(A,B,C),(C,D),(D,E),(D,F)$ has greater probability when compared to $(A,B),(B,C),(CD),(C,D,E),(D,F)$. In addition we can remark that the arrangement $(A,B),(B,C),(C,D),(D,E),(D,F)$ has the least probability in each case.

Figure 4.13 contrast the cubic and higher order polynomial approximations derived for estimating the probability of partial and full information awareness for respectively clustering arrangement $(A,B,C),(C,D),(D,E),(D,F)$ and $(A,B),(B,C),(CD),(C,D,E),(D,F)$ using the corresponding data points from Figure 4.12. We can see that for $(A,B,C),(C,D),(D,E),(D,F)$, the cubic approximation is not fitting well the data points and has a coefficient of determination value of 0.9834. For $(A,B),(B,C),(CD),(C,D,E),(D,F)$, we can see that the cubic approximation is fitting better with a coefficient of determination value of 0.9945. However, in both cases, the cubic approximation is unusable for small fwr values since it provides negative probability values. In contrast, a sixth and respectively fifth degree polynomial mostly avoid this issue and fit better the data points of $(A,B,C),(C,D),(D,E),(D,F)$ and $(A,B),(B,C),(CD),(C,D,E),(D,F)$. This shows that no simple approximation can be used for characterizing the information diffusion. The dynamics is affected by the hierarchical relationships among nodes rather than just cluster structure and can be further impacted by dissimilar cluster disruption levels.

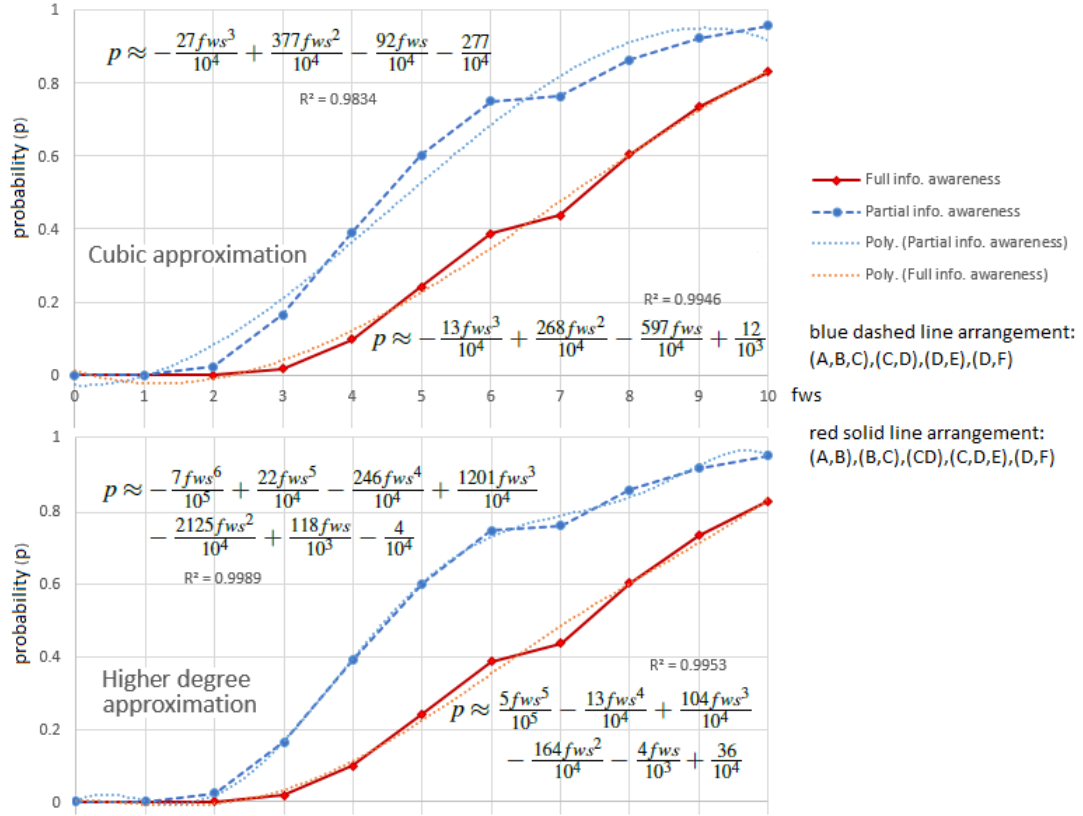


Figure 4.13: Comparison of cubic and higher degree polynomial approximations of shared information awareness for two cluster arrangements

$probability > 0.5$	0.602	0.748	0.762	0.862	0.922	0.956
fws	5	<u>6</u>	7	8	9	10
$u=probability/fws$	0.120	<u>0.124</u>	0.108	0.107	0.102	0.095

Table 4.1: Identification of best fresh window size relative to acceptable probability values over a threshold of 50%

The foregoing probabilistic model-checking results can also enable the identification of the best fresh window size for a particular clustering arrangement and disruption level relative to an acceptable threshold for the probability value. More specifically, this typically involves the optimization (e.g., maximization) of a related utility function. For example, we may be satisfied if the probability value for achieving information awareness exceeds a certain threshold but would like to have a small window in order to obtain the information with less delay. Thus, a simple utility function can be represented for instance by the ratio of the probability value to the size of the fresh

window: $u = \text{probability}/fws$. In this context, Table 4.1 provides the *probability* values greater than a threshold of 50% for achieving partial information awareness along with the corresponding fresh window size (fws) and the value of the utility function (u) for the clustering arrangement (A,B,C),(C,D),(D,E),(D,F), based on the corresponding graph in Figure 4.12. The best trade-off, which maximizes the utility function, can be observed for $fws = 6$ where $u = 0.748/6 \approx 0.124$.

4.7 Summary

This chapter presented a protocol for achieving high level of shared information awareness in the context of distributed and hierarchy-aware monitoring of plan execution in environments with noisy communication. The information sharing mechanism has been assessed via probabilistic model-checking for different combinations of clustering arrangements and communication disruption levels. A case study was also presented to illustrate the assessment of probabilistic properties using PRISM model-checker. The obtained results show that the protocol exhibits resilience to high levels of communication errors rates as long as there is enough scope for mitigation through retransmission. In addition, the resilience characteristic represents an instrumental aspect in supporting distributed monitoring along with the ability of the participants to join or leave a particular cluster.

The contribution resides in the following. First, a hierarchy-aware distributed monitoring approach was elaborated based on asymmetric clustering of collaborating participants executing a plan in communication disrupted environment. Second, the information sharing mechanism among hierarchically related participants was formalized using communicating Markov Decision Processes, which can be subjected to probabilistic model-checking. Third, the assessment of the degree of shared information awareness was demonstrated based on verifying formal specifications expressed in probabilistic temporal logic. Fourth, it was shown how model-checking results can be used in order to maximize an information awareness utility function. In addition, the proposed approach leverages hierarchical information gathering and forwarding. In this context, it employs the concept of asymmetric clustering of participants along with the gossiping paradigm, for collaborative plan execution monitoring in communication disrupted environment, where the participant's rank determines the information received.

Compared to [135], which employs symbolic model-checking to find resources in grid computing environments, the approach in this chapter employs probabilistic model checking. This allows to assess probabilistic behaviours via properties expressed in probabilistic temporal logic, with enhanced expressiveness. With respect to [115], which studies mobile ad hoc networks (MANETs) routing protocols and highlights the benefits of hierarchical routing strategies, the approach in this chapter also leverages hierarchical information gathering and forwarding. In contrast to [31], which adopts Hierarchical Task Network (HTN) to optimize tasks for instructional designers in Computer Supported Collaborative Learning (CSCL), based on units of learning, the approach in this chapter focuses on hierarchical sharing of knowledge. However, event based units of information are considered as part of monitoring a plan being executed.

The limitations result from the inherent limitations of model-checking concerning the problem of state explosion. However, for models with moderate state space, model-checking allows to conduct a thorough assessment of model properties. The obtained results can provide useful insights with respect to the underlying dynamics of interest. In addition, model-checkers such as PRISM allow to assess properties with approximation of various degrees of accuracy via statistical model-checking or simulation when the state space is too large for thorough verification. Moreover, the proposed hierarchy-aware information sharing paradigm is tailored for plan execution monitoring taking place in hostile environments and is less suited for environments where communication disruptions are sparsely encountered.

The advantages of the proposed approach in the context of collaborative monitoring of plan execution consist in providing capabilities for assessing the degree of shared information awareness, according to a chain of command. This includes event information typically involving plan deviations encountered in different cluster arrangements and under various disruption levels. In this respect, the concept of asymmetrical clustering was employed to capture the hierarchical relationships among the participants, according to their rank. A notable area of applicability is related to adaptive planning in response to changing circumstances during collaborative plan execution taking place in hostile environments prone to communication disruption.

Chapter 5

Efficient Sensor Network Management for Asset Localization

Asset localization represents a key application of wireless sensor network (WSN) with specific relevance for transport network monitoring and plan execution. This chapter is addressing the problem of WSN management where resource constrained sensors, in terms of capacity, sensing range and energy, are assigned to multiple target assets in order to localize the assets with minimized error. We consider a heterogeneous network of omnidirectional sensors, each having an individual capacity to focus on a number of targets and a specific range to accurately estimate its distances to the targets that it is focusing on. A proper localization of each target requires a minimum of K (typically three) sensors where the target location area is derived using the intersection of the K range circles. The problem is further analysed under the constraint of an overall WSN energy budget, which limits the possible assignments for the capacitated sensors. Restricting the energy budget leads to a trade-off between energy conservation and localization performance.

In this context, a heuristic solution approach is proposed, leveraging evolutionary learning followed by meta-heuristic improvements based on target swapping among sensors. The proposed approach involves the minimization of a quantifier composed of the total localization area for all target assets along with a penalty for each target assigned less than minimum sensors. An instructive case study is used to illustrate the approach while its effectiveness is assessed experimentally via benchmark results over on a data-set derived from a known vehicle routing problem set.

5.1 Context and Assumptions

Logistics planning in situations of crisis involves designing and carrying out tasks such as: movement of assets (vehicles, goods, etc.), stationing assets at specific deploy points, etc. over the available supply chain networks. Characteristically, those preplanned tasks are often executed over perilous territory in hostile environments. In such an environment, asset localization represents an important aspect due to high likelihood of potential plan deviation due to unexpected and unaccounted events. A timely assessment of such deviations may provide a cutting edge in pursuit of successful and efficient completion of planned tasks. In this setting, we address the issue of asset localization under the assumption that the tasked assets may be unable to report their own location, being required to maintain radio silence in hostile environments. Thus, we consider an external set of heterogeneous sensors that can be used to perform asset localization. In this context of multisensor-multitarget applications such as asset localization, the core issue is the association of information from multiple sensors in order to support target identification [83]. The underlying problem is discussed in the general class of multidimensional assignment problems [20, 113, 138]. We aim at elaborating a framework for multisensor-multitarget asset localization over an instrumented space containing deployed sensors.

The multisensor-multitarget assignment problem represents an extension of the focus of attention (FOA) problem [60] where each target is assigned to only a pair of sensors while each sensor, within its range, has the capacity of focusing on only one target. We address a generalized problem where omnidirectional sensors have increased capacities while the targets require a minimum number of sensors (3 or more for appropriate localization). The generalized problem has NP-hard complexity since it extends the FOA problem. FOA is generally NP-hard, as it is a specific case of the multi-dimensional assignment problem [147]. In this context, we propose an evolutionary learning-based heuristic aimed at finding the best sensor-target assignment combination that would minimize the overall localization error and penalty levels (when the sensor network is under-capacitated). The proposed approach allows to effectively employ heterogeneous sensor networks where various sensors may have different capacities and sensing ranges.

5.2 Problem Description and Modelling

The problem of asset localization through WSN is composed of two sub-problems: A sensor assignment problem for each target (typically referred as the sensor focus of attention problem) and a localization problem for each target (deriving location of each target from a particular assignment of sensors to targets). The latter problem is approached by triangulation since we consider omnidirectional sensors with different sensing ranges and target focusing capacity. Thus, we consider that at least three different sensors need to be assigned to a target asset in order to properly localize it. In this setting, we aim at finding the most suitable multisensor-multitarget assignment that would result in minimized overall localization error.

5.2.1 Sensor Focus of Attention

Sensor focus of attention (FOA) and related sensor assignment problems provide key modelling concepts for finding optimized configurations over a sensor network, particularly in the case where heterogeneous sensors can focus in a given time unit only on a subset of the targets in their coverage area. In this context, distinct sensors may have different abilities in terms of detection range (e.g., due to difference in elevation level) and target focusing capacity (i.e., one or more targets per time unit). For example, more capable sensors are often placed in locations deemed of higher importance (e.g., higher trafficability potential). An effective sensor to target assignment minimizes the localization error for the targets by appropriately assigning sensors to targets. The heterogeneity aspect introduces an extended level of complexity due to the vast number of possible sensor assignment combinations. This requires heuristic search methods for near-optimal sensor assignment. However, since the localization error cost can be trivially reduced by not assigning sensors to some targets, such solutions can be deemed as being invalid or otherwise can be considered along with a sufficiently large penalty to render such solutions very unattractive.

5.2.2 Problem Statement

We investigate the problem of asset/target localization as an application over WSN. Over an instrumented space, at a high level of abstraction, a sensor network can be represented by a bipartite graph $G_{SM} = \langle S, M, E_{SM} \rangle$ where S and M denote n sensors and m targets respectively.

E_{SM} denotes a set of directed edges where each edge originates from an element of S to an element of M indicating a possible assignment of a sensor to a target. Given that a sensor node i and a target node j have defined locations, d_{ij} is considered as estimated distance with respect to target j . In this setting, let a set of variables, x_{ij} , $i \in S$, $j \in M$, determine the assignment of sensors to the targets. As such, if $x_{ij} = 1$, an assignment exists between sensor i and target j . In the localization problem, a collection of sensors assigned to a target determine the potential location of the target within an intersection area. The latter is desired to be as small as possible. However, given a limited number of sensors with restricted abilities, such an optimal assignment of sensors to targets involves a non-linear optimization problem constrained from the aforementioned limitations. In this context, the problem is modelled with the following objective function and constraints:

$$\min \sum_{j=1}^m f_j(x_{1j}, \dots, x_{ij}, \dots, x_{nj}) \quad (5.1)$$

Subject to:

$$\sum_{j=1}^m x_{ij} \leq C_i, \quad \forall i \in N \quad (5.2)$$

$$\sum_{i=1}^n x_{ij} \geq K \quad \forall j \in M \quad (5.3)$$

$$d_{ij} \times x_{ij} \leq R_i \quad \forall i \in N, j \in M \quad (5.4)$$

$$x_{ij} \in \{0, 1\} \quad (5.5)$$

Eqn. (5.1) is the objective function where f_j determines an area around every target j as per its sensor assignments. Computing such an area, in general case, is a complex mathematical problem [90]. However, in the context of a solution finding algorithm, the values can be considered as pre-computed on demand and retrieved using memory-efficient data structures. Eqn. (5.2) states that a sensor cannot be assigned more than its capacity of focusing on C_i targets. Similarly, Eqn. (5.3) puts a constraint that every target should be assigned to at least K sensors. Eqn. (5.4) assures that a sensor i can be assigned to target j only if the distance is within the specified coverage or monitoring range of the sensor, R_i . The aforementioned model presents two challenges. First, it requires an equivalent linear model to assess the suitability of applying heuristic solution generation techniques. Second, some problem instances may have no feasible solutions for particular K

values. This relates to the one-dimensional bin-packing problem, which has *NP*-hard computation complexity. This can be addressed by relaxing Eqn. (5.3) as will be detailed in the sequel.

In order to address the first challenge, we rewrite the aforementioned model in a linear form using higher-dimension decision variables. Let $x_{i_1, \dots, i_n j}$ be an assignment of all sensors ($i_1 \dots i_n$) to target j , which yields a localization area $c_{i_1, \dots, i_n j}$ around target j . Precisely, $x_{i_1, \dots, i_n j}$ is a multi-dimensional binary decision variable. Then, the following model captures the aforementioned equations, Eqns. (5.1)-(5.4), in linear programming:

$$\min \sum_{j=1}^m \sum_{i_1=0}^1 \cdots \sum_{i_n=0}^1 c_{i_1, \dots, i_n j} \cdot x_{i_1, \dots, i_n j} \quad (5.6)$$

Subject to:

$$x_{0, \dots, 0 j} = 0, \quad \forall j \in \{1, \dots, m\} \quad (5.7)$$

$$\sum_{i_1=0}^1 \cdots \sum_{i_n=0}^1 x_{i_1, \dots, i_n j} = 1, \quad \forall j \in \{1, \dots, m\} \quad (5.8)$$

$$\sum_{j=1}^m \sum_{i_1=0}^1 \cdots \sum_{i_{p-1}=0}^1 \sum_{i_{p+1}=0}^1 \cdots \sum_{i_n=0}^1 x_{i_1, \dots, i_{p-1}, 1, i_{p+1}, \dots, i_n j} \leq C_{i_p}, \quad \forall p \in \{i_1, \dots, i_n\} \quad (5.9)$$

$$d_{i_p j} \left(\sum_{i_1=0}^1 \cdots \sum_{i_{p-1}=0}^1 \sum_{i_{p+1}=0}^1 \cdots \sum_{i_n=0}^1 x_{i_1, \dots, i_{p-1}, 1, i_{p+1}, \dots, i_n j} \right) \leq R_{i_p}, \quad \forall p \in \{i_1, \dots, i_n\}, j \in \{1, \dots, m\} \quad (5.10)$$

$$\begin{aligned} & \sum_{i_2=0}^1 \cdots \sum_{i_n=0}^1 x_{i_1, i_2, \dots, i_n j} + \cdots + \sum_{i_1=0}^1 \cdots \sum_{i_{p-1}=0}^1 \sum_{i_{p+1}=0}^1 \cdots \sum_{i_n=0}^1 x_{i_1, \dots, i_{p-1}, 1, i_{p+1}, \dots, i_n j} + \\ & \cdots + \sum_{i_1=0}^1 \cdots \sum_{i_{n-1}=0}^1 x_{i_1, \dots, i_{n-1}, 1 j} \geq K, \forall j \in \{1, \dots, m\} \end{aligned} \quad (5.11)$$

$$x_{i_1, \dots, i_n} \in \{0, 1\} \quad (5.12)$$

In this model, $x_{i_1, \dots, i_n j}$ is an n -dimensional decision variable where i_p indicates participation of sensor p to localize target j . If $x_{i_1, \dots, i_n j}$ is 1, the allocation of a subset of n sensors get established for target j with all participating and non-participating sensors to locate target j . Accordingly, $c_{i_1, \dots, i_n j}$ indicates the allocation cost in similar manner to f_j function in Eqn. (5.1). Eqns. (5.7) and (5.8) assure that every target is assigned to at least a set of sensors. Eqn. (5.9) specifies that sensor p is allocated to no more then C_{i_p} targets. Eqn. (5.10) assures that a sensor can be only assigned to a target when the target is within its monitoring range. Eqn. (5.11) requires allocation of at least

K sensors per target. Eqn. (5.12) is used to define the scope of the n -dimensional variable x_{i_1, \dots, i_n} .

In order to address the second challenge of infeasible solution, we consider an objective function that minimizes localization area along with a related penalty. This model captures a capacity restricted sensor network where the total number of assignments is no more than a predetermined budget B . When B is less than $|M| \times K$, this means that Eqn. (5.3) is not satisfied for at least one target. Under the assumption that each sensor requires one unit of energy to focus on one target, we consider B as the energy budget since it is proportional to the total use of energy by the sensors to engage in the process of localization. The imposed penalty per target is proportional to the lack of allocated sensor(s) with respect to the requirement (K).

We relax Eqn. (5.3) and modify the objective function in Eqn. (5.1) by including a penalty for every target where less than K sensors are assigned, in either additive (Eqn. (5.13)) or multiplicative ((Eqn. (5.14)) form. In Eqn. (5.13), we note the introduction of a base factor ρ_1 multiplied by the number of missing sensors, where ρ_1 is sufficiently large. In addition, with respect to both Eqn. (5.13) and Eqn. (5.14), $f_j(x_{1j}, \dots, x_{ij}, \dots, x_{nj})$ is expected to represent a correspondingly larger area due to lack of adequate sensors, including the case where a target has no sensors assigned.

$$\min \sum_{j=1}^m \left[f_j(x_{1j}, \dots, x_{ij}, \dots, x_{nj}) + \max\left(0, \rho_1 \times \left(K - \sum_{i=1}^n x_{ij}\right)\right) \right] \quad (5.13)$$

$$\min \sum_{j=1}^m \left[f_j(x_{1j}, \dots, x_{ij}, \dots, x_{nj}) \times \max\left(1, \left(K + 1 - \sum_{i=1}^n x_{ij}\right)\right) \right] \quad (5.14)$$

Similar to the previous discussed models, the objective function is subjected to the constraints in Eqns. (5.2), (5.4) and (5.5) along with an additional energy budget constraint presented in Eqn. (5.15). The latter can also be modified to include it in the linear model in the same manner as Eqn. (5.4) was modified into to Eqn. (5.11).

$$\sum_{i=1}^n \sum_{j=1}^m x_{ij} \leq B \quad B < |M| \times K \quad (5.15)$$

This model is used as a guide for the performance of the proposed approach for the under-capacitated and energy budget restricted problem instances.

5.3 Proposed Approach

In this section, we describe the proposed solution approach with emphasis on localization error reduction. The aim is to elaborate advanced capabilities for an advisory asset localization system that decision makers can use in the context of logistics and supply chain management.

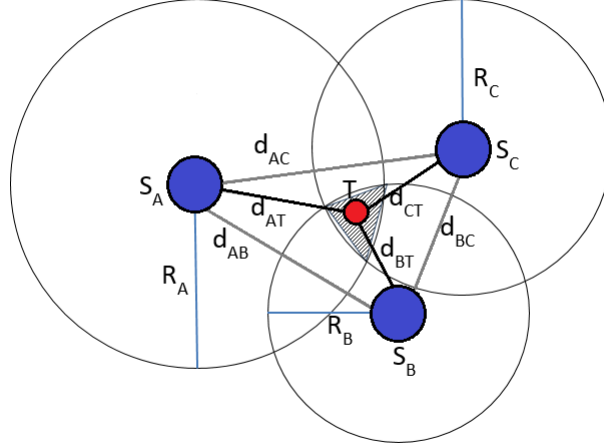


Figure 5.1: Target T inside the triangle delimited by $\{S_A, S_B, S_C\}$ and marked intersection area for possible error in target identification

In the absence of information exchange with the targets, asset localization can be carried out using sensor data in relation to specific target signatures. In this respect, one may assume that the available sensors are able to discriminate a target signature, based for example on generated heat, reflected electromagnetic waves, sound, vibration, etc. Each sensor having a specific location and estimated distance from a target within its detecting range, can participate along with other sensors, that can detect the same target, in localizing the target with certain error corresponding to an intersection area. The assignment of at least K (typically 3) sensors to a target, establishes a possible area [90] where the target can be situated¹, as seen in Figure 5.1.

Since the problem of multi-sensor to multi-target assignment exhibits combinatorial explosion, a heuristic solution generation approach is proposed whereby near-optimal solutions can be obtained in a computationally cost effective manner. The heuristic solution search involves the exploration of various sensor-target assignments in order to identify progressively better solutions.

¹A polygon-based approximation can be found at: <https://bit.ly/2YeJz5z>, Last accessed on 1 March 2019.

This requires the means to compare solutions that may fail to satisfy certain constraints such as assigning at least K sensors to each target or respecting the energy budget restriction.

In order to address these situations, assigning less than K sensors to a target is allowed as well as exceeding the energy budget restriction but a composed penalty is imposed. Thus, we relax Eqns. (5.4) and (5.15) while modifying the objective function in Eqn. (5.1) by including a target specific penalty whenever less than K sensors are assigned along with an overall penalty on the whole solution in the case where the energy budget restriction is exceeded. The overall penalty increases the solution cost by taking into account the increase over the energy budget restriction. According to the penalty type there are two variants corresponding to either additive (Eqn. (5.16)) or multiplicative (Eqn. (5.17)) form. In Eqn. (5.16), we can note the presence of a supplementary base factor ρ_2 multiplied with the value exceeding the energy budget restriction, where ρ_2 is sufficiently large.

$$\min \sum_{j=1}^m \left[f_j(x_{1j}, \dots, x_{ij}, \dots, x_{nj}) + \max\left(0, \rho_1 \times \left(K - \sum_{i=1}^n x_{ij}\right)\right) \right] + \max\left(0, \rho_2 \times \left(\sum_{j=1}^m \sum_{i=1}^n x_{ij} - B\right)\right) \quad (5.16)$$

$$\min \sum_{j=1}^m \left[f_j(x_{1j}, \dots, x_{ij}, \dots, x_{nj}) \times \max\left(1, \left(K + 1 - \sum_{i=1}^n x_{ij}\right)\right) \right] \times \max\left(1, \frac{\sum_{j=1}^m \sum_{i=1}^n x_{ij}}{B}\right) \quad (5.17)$$

The key aspects of the proposed approach are described next. In the context of solving challenging optimization problems, heuristic techniques are typically employed to obtain near-optimal solutions. The general idea of the heuristic solution generation technique is conveyed by Algorithm 3 along with a subsequent discussion. The proposed approach is inspired by the learning-based evolutionary concept described in [131], which employs an evolving population of solutions in the context of partitioning a set of customers among the depots of a supply chain in order to minimize overall routing cost. However, the scope of our problem is notably more challenging since it involves the underlying problem of multi-dimensional assignment with difficult constraints. These involve the assignment of multiple sensors to multiple targets, taking into account that each sensor can be assigned to any number of targets in its range, according to its capacity. Thus, in our case, we iteratively spawn successive generations of solutions, produced using a pseudo random number choice generator that allows to obtain diverse multisensor-multitarget assignment combinations according to the available sensor assignment choices (initially derived

Algorithm 3 Heuristic Search (HS) for near-optimal multisensor-multitarget assignment

```
1: Input: max_iteration(maxIter), remove_count(rCnt), sample_count(sCnt), elite_count(eCnt), seed;  
2: Global Knowledge: SensorSet(sSet), TargetSet(tSet);  
3: Initially: crtBestSol  $\leftarrow \{\}$ , rndGen  $\leftarrow$  PseudoRandomGenerator(seed);  
4: while (maxIter > 0) do  
5:   if GetMaxCombinationCount(sSet, tSet)  $\leq$  sCnt then  
6:     allSensorTargetCombinations  $\leftarrow$  GenerateAllSensorTargetCombinations(sSet, tSet);  
7:     for each sensorTargetCombination in allSensorTargetCombinations do  
8:       if Cost(sensorTargetCombination) < Cost(crtBestSol) or crtBestSol =  $\{\}$  then  
9:         crtBestSol  $\leftarrow \{ \text{Cost}(\text{sensorTargetCombination}), \text{sensorTargetCombination} \}$ ;  
10:      end if  
11:    end for  
12:    break;  
13:  else  
14:    sortedSearchMap =  $\{\}$ ;  
15:    for i=0 to sCnt do  
16:      exploreSol  $\leftarrow$  GeneratePseudoRandomSensorTargetCombination(sSet, tSet, rndGen);  
17:      if Size(sortedSearchMap) < eCnt or Cost(exploreSol) < LastKey(sortedSearchMap) then  
18:        Insert {Cost(exploreSol), exploreSol} into sortedSearchMap;  
19:        if Size(sortedSearchMap) > eCnt then  
20:          remove LastEntry(sortedSearchMap) from sortedSearchMap  
21:        end if  
22:      end if  
23:    end for  
24:    end if  
25:    if crtBestSol =  $\{\}$  or Cost(crtBestSol) > FirstKey(sortedSearchMap) then  
26:      crtBestSol  $\leftarrow$  FirstEntry(sortedSearchMap);  
27:    end if  
28:    sensorCombinationVoteMap  $\leftarrow$  CountAndRankPerSensorCombinationVotes(sortedSearchMap);  
29:    srCnt  $\leftarrow$  rCnt;  
30:    for each {sensor, combination} in sensorCombinationVoteMap do  
31:      if srCnt = 0 then  
32:        break;  
33:      else  
34:        if crtBestSol does not contain combination and CombinationChoiceCount(sensor) > 1 then  
35:          remove combination choice from sensor  
36:          srCnt  $\leftarrow$  srCnt - 1;  
37:        end if  
38:      end if  
39:    end for  
40:    maxIter  $\leftarrow$  maxIter - 1;  
41:  end while  
42: return crtBestSol;
```

from the sensing ranges and capacities of the sensors). Then, the cost and assignment combinations of the solutions generated in a given iteration allows to learn the sensor assignment choices that are cost-wise ineffective in order to remove one or more at every iteration. At the limit, the total number of available assignment choices would steadily decrease until each sensor would be left with only one assignment choice, corresponding to the final solution.

In the context of the algorithm, the population size of each generation represents an input parameter and each individual multisensor-multitarget combination in a generation has a corresponding cost, which is used to rank the individual combinations in each generation. From all individuals in a generation, only an elite number is retained where the number of elite solution represents another input parameter. The current best solution is updated in each iteration whenever a better solution is identified in the current iteration elite solutions. Moreover, the elite solutions are analysed based on a voting scheme over the sensor-target assignment combination in order to rank the cost effectiveness of the underlying target assignment combinations across the elite solutions. Subsequently, from the combinations with the least potential to participate in good quality (lower cost) solutions, a certain number (another input parameter) is selected to be removed from the possible assignment choices of their respective sensors, before spawning the next generation. The removal of certain sensor-target assignment combination choices from various sensors, after each iteration, reduces the solution search space and allows the procedure to converge to a near-optimal solution over a number (stop condition parameter) of successive iterations or until the remaining combinations are less than or equal to the population size parameter value. In this case, an exhaustive search provides the most cost effective solution in the remaining solution search space.

In the following, we detail the key points of Algorithm 3 from a higher level of abstraction in order to convey more effectively the underlying concept. The search procedure is initialized in line 1 with the following input parameters:

- $\text{max_iteration}(\text{maxIter})$: maximum number of generations the procedure can spawn over successive iterations (stop condition parameter).
- $\text{remove_count}(\text{rCnt})$: number of assignment combination choices that can be removed per iteration.
- $\text{sample_count}(\text{sCnt})$: population size to be sampled from the search space at each iteration.

- *elite_count(eCnt)*: number of elite solutions to be retained from the sampled population on each iteration.
- *seed*: value to initialize the random number choice generator used for solution sampling.

At line 2, we can see that the sensor set (*sSet*) and the target set (*tSet*) are considered as global knowledge. Furthermore, at line 3, the current best solution (*crtBestSol*) is set initially empty while the random number generator (*rndGen*) is initialized with the *seed* value.

The procedure continues at line 4 with a while loop testing whether the maximum number of iterations has been reached. In the while loop, the procedure evaluates at line 5 whether the maximum number of combinations for the sensor set and target set is less than *sCnt*. In this case, an exhaustive search is used to identify the final solution before terminating the procedure (lines 6 to 12). Otherwise, at line 14, a cost-wise sorted (in ascending manner) solution search map (*sortedSearchMap*) is set to empty and a for loop over the sample count is used to explore pseudo randomly generated solutions, aggregating the elite qualifying ones in the *sortedSearchMap* (lines 15 to 23). Then, (*crtBestSol*) is updated if needed (lines 25 to 27). At line 28, a sorted sensor combination vote map (*sensorCombinationVoteMap*) is used to hold the count and rank corresponding to the per sensor combination votes obtained by analysing the elite solutions stored in the *sortedSearchMap*. The *sensorCombinationVoteMap* is sorted in a manner that allows iterating over its elements from the least likely combinations to participate in competitive elite solutions to the most likely. Thus, a number of sensor assignment combination choices equal to *rCnt* is removed from their respective sensors by iterating over the *sensorCombinationVoteMap* (lines 29 to 39). The value of *maxIter* is decremented at line 40 and after completing the while loop, the procedure returns *crtBestSol* (at line 42).

5.4 Case Study

We present next an application of the proposed approach in the context of an illustrative case study problem depicted in Figure 5.2 (left). The problem involves a sensor network comprising 7 sensors (a, b, c, d, e, f, g) that have to localize 4 targets (P, Q, R, S) in their coverage area. The details of the problem are shown in Table 5.1, which provides for each sensor the corresponding range, target focusing capacity and the targets in its range. We note that there are

multiple possible assignments for each sensor ranging from 2 (sensor c), 3 (sensors a, b and e) and 6 (sensors d, f and g) yielding a total of 11664 possible assignments. Also, each sensor has a tolerance of 5% for target distance estimation. Figure 5.2 (middle) presents the solution obtained with a nearest neighbour sensor assignment while Figure 5.2 (right) presents the optimal solution.

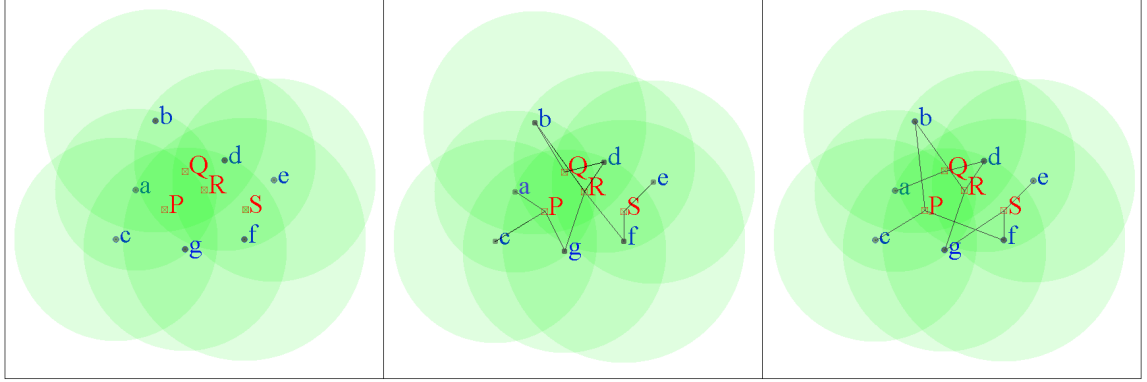


Figure 5.2: Case study problem (left), nearest neighbour solution cost=2742.1184 (middle), optimal solution cost=392.1358 (right)

Sensor	a	b	c	d	e	f	g
Range	80	110	100	90	100	120	100
Target focusing capacity	1	2	1	2	1	2	2
Targets in range	P, Q, R	P, Q, R	P, Q	P, Q, R, S	Q, R, S	P, Q, R, S	P, Q, R, S
Target assignment combinations	3	3	2	6	3	6	6

Table 5.1: Asset localization case study problem details

Given that we aim for at least three sensors assigned for each target, the best solution will include as much as possible corresponding sensor assignments. However, the case study problem data is chosen such that any valid solution will have at least one target with an assignment of less than 3 sensors. This allows to highlight that in the general case, there is no advantage to prune the combinations where less than 3 sensors are assigned per target since such assignment can be encountered even in the optimal solution. In this context, a target related penalty is applied when only one or two sensors are assigned for a target. In the absence of such penalty, certain targets can have comparably large localization error cost while others can have notably smaller localization error cost (the overall localization error cost may be reduced but some targets will be localized very poorly). The lowest penalty is applied when only two sensors are assigned for a target and

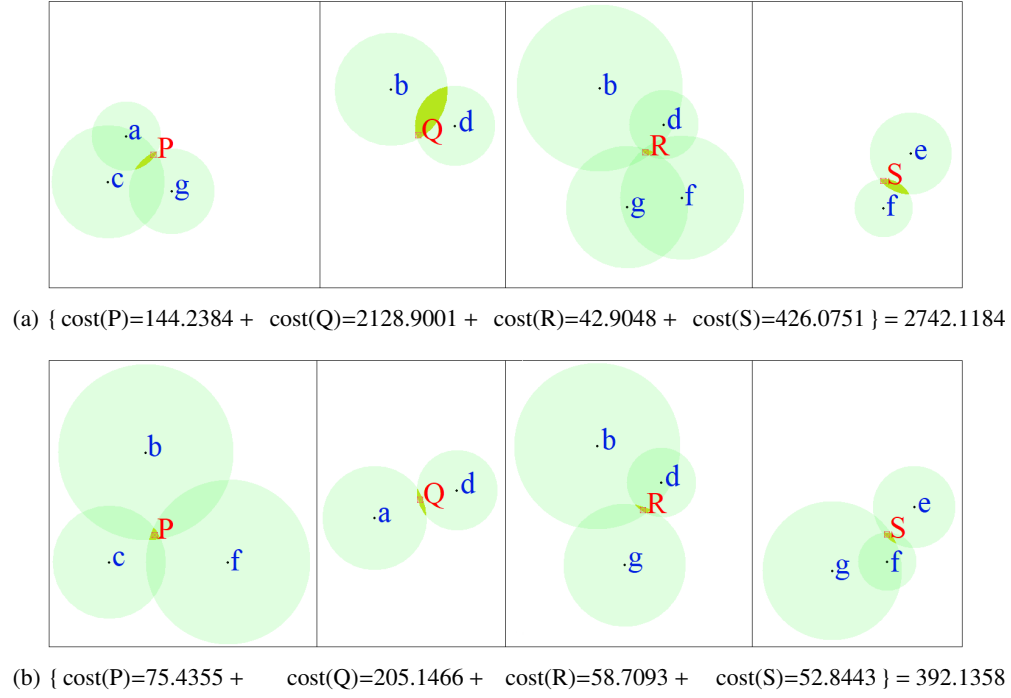


Figure 5.3: Breakdown cost comparison between the nearest neighbour solution (a) and the optimal solution (b)

a higher (double) penalty is applied when only one sensor is assigned for a particular target since such assignment yields higher localization error cost. Thus, the penalty varies corresponding to the number of missing sensors and can be additive for larger problems or multiplicative for smaller problems such as the one presented in the case study.

In Figure 5.3, we contrast the nearest neighbour solution obtained by assigning the sensors to the closest targets in their range against the heuristically obtained optimal solution (confirmed via exhaustive search) and provide the cost breakdown comparison per target. The sensor assignment for each target corresponding to the nearest neighbour solution is: P: (a, c, g), Q: (b, d), R: (b, d, g, f), S: (e, f) while the assignment for the optimal solution is: P: (b, c, f), Q: (a, d), R: (b, d, g), S: (e, f, g). We note that with the exception of target R, which has a slightly lower cost in the nearest neighbour solution, all other targets exhibit notably smaller cost in the optimal solution compared to the nearest neighbour solution. The latter has a lower cost for target R: (b, d, g, f) assigning it 4 sensors while assigning only 2 sensors to targets Q: (b, d) and S: (e, f) thereby incurring notably higher total penalty compared to the

optimal solution, which has only one assignment with penalty for Q: (a, d). Even if we disregard the penalty, the cost for target Q ($205.1466/2=102.5733$) is markedly lower in the optimal solution compared to the cost for target Q ($2128.9001/2=1064.4500$) in the nearest neighbour solution.

We illustrate next the heuristic solution generation for a sample count of 40 and an elite set size of maximum 10 solutions.

Table 5.2 presents the sensor-target assignment combinations, sorted column-wise and row-wise as follows: column-wise, by minimum received votes (primary criterion) and by the difference between maximum and minimum votes (secondary criterion); row-wise, by vote count (primary criterion) and by participating average solution cost (secondary criterion). Accordingly, we can see in Table 5.2 the breakdown of the received votes per sensor, ordered in increasing order of the received votes. This ordering allows to identify the combinations with the least potential to participate in competitive solutions.

For the cases where the same number of minimum votes are received by different sensor combinations, the difference between the maximum and the minimum votes (of the same sensor combinations) is used as sorting criterion. Thus, the sensors with a larger difference are considered before the ones with a smaller difference. This allows to segregate among different combinations receiving the least amount of votes. Thus, a larger difference (for the combinations of a particular sensor) corresponds to having a higher confidence that the combinations receiving minimum votes (for that sensor) are less likely to be part of competitive solutions. This contrasts to other sensor combinations (receiving the same amount of minimum votes) for which the difference is smaller.

When two sensors have the same number of minimum votes and the same difference between their maximum and minimum votes (received by their respective combinations), then the corresponding minimum votes combinations have similarly low competitive potential. For instance, the first and second columns of Table 5.2 show that sensors d and g have combinations

Sensors: d	g	f	a	b	e	c
[Q,S]/5/2785.4592	[P,S]/4/5112.416	[P,S]/3/4941.793	[R]/4/2416.4458	[P,R]/4/3261.83	[S]/4/5006.1055	[Q]/6/4231.99
[Q,R]/2/6123.157	[P,R]/2/3387.3562	[Q,S]/2/1188.0356	[P]/4/6627.453	[P,Q]/3/2837.9312	[R]/3/2526.526	[P]/4/4136.799
[R,S]/2/7097.16	[P,Q]/2/3505.8823	[R,S]/2/3644.856	[Q]/2/2881.7705	[Q,R]/3/6792.675	[Q]/3/4778.38	
[P,R]/1/1571.2064	[Q,S]/2/3851.498	[Q,R]/1/5126.551				
[P,Q]/0/0.0	[Q,R]/0/0.0	[P,Q]/1/6114.5254				
[P,S]/0/0.0	[R,S]/0/0.0	[P,R]/1/6206.9				
min votes: 0	0	1	2	3	3	4

Table 5.2: Sensor-target assignment combinations and breakdown of the received votes

Elite solution cost	Sensor assignment			
636.9903	P=[b; f; g]	Q =[a; c; d]	R=[b; e; g]	S =[d ; f]
804.86487	P=[b; c; g]	Q =[b; d ; f; g]	R=[a; e]	S =[d ; f]
1151.9899	P=[b; g]	Q =[c; d ; e]	R=[a; b; f]	S =[d ; f; g]
1571.2064	P=[b; c; d]	Q=[b; f; g]	R=[a; d]	S=[e; f; g]
5126.551	P=[b; g]	Q =[a; c; d ; e; f]	R=[b; f]	S =[d ; g]
6114.5254	P=[a; c; f; g]	Q=[b; d; f]	R=[b; d]	S=[e; g]
6131.7896	P=[a; b; f]	Q=[c; d; g]	R=[b; d]	S=[e; f; g]
6137.722	P=[b; g]	Q=[b; c]	R=[a; d; e; f; g]	S=[d; f]
6206.9	P=[a; f; g]	Q =[b; c; d ; g]	R=[b; f]	S =[d ; e]
8056.5986	P=[a; c; f; g]	Q=[b; e]	R=[b; d]	S=[d; f; g]

Table 5.3: Elite solutions for the first iteration, with most amount of votes (5) provided by 5 solutions for combination [Q, S] of sensor d

receiving a minimum number of votes equals to 0 ([P,Q] and [P,S] for d, respectively [Q,R] and [R,S] for g. However, the maximum number of votes received by d is 5 for combination [Q,S] and respectively 4 for sensor d combination [P,S]. Since the difference between maximum and minimum number of votes is higher for d, the latter appears before g. The combinations belonging to each sensor are also sorted in decreasing order of their received votes as shown in Table 5.2. When some combinations belonging to the same sensor receive the same amount of votes, the average cost of the participating solutions is taken as supplementary sorting criterion. This favours the combinations that are cost-wise beneficial across multiple solutions. For example, according to Table 5.2, sensor b receives 4 votes for combination [P,R] (with average cost of participating solutions of 3261.83) and 3 votes for [P,Q] and [Q,R] (with average cost of participating solutions of 2837.9312 and 6792.675 respectively). In this example, [P,R] appears first since it has more votes, while [P,Q] appears before [Q,R] since it has lower average cost for the same votes.

The potential candidate combinations to be pruned, can be identified from Table 5.2, before the next sampling step. The first candidate is [P, S] for sensor d (marked in bold face). The second candidate is [P, Q] also for sensor d. The third candidate is [R, S] for sensor g. The fourth candidate is [Q, R] also for sensor g. The fifth candidate is [P, R] for sensor d rather than [P, R] for sensor f since the difference between the maximum and minimum votes for sensor d is greater than the one for sensor f. Further candidates can similarly be identified. Table 5.3 provides the top 10 elite solutions obtained in the first iteration along with the most amount of votes (5) provided by 5 of the solutions (with average cost of $\frac{636.9903+804.86487+1151.9899+5126.551+6206.9}{5} = 2785.4592$)

for target combination [Q, S] of sensor d (targets and sensor marked in bold face). In contrast, the least amount of votes (0) have been received in the first iteration by target combinations [P, Q] and [P, S] for sensor d and [Q, R] and [R, S] for sensor g since these combinations do not appear across all of the elite solutions, thus receiving no votes.

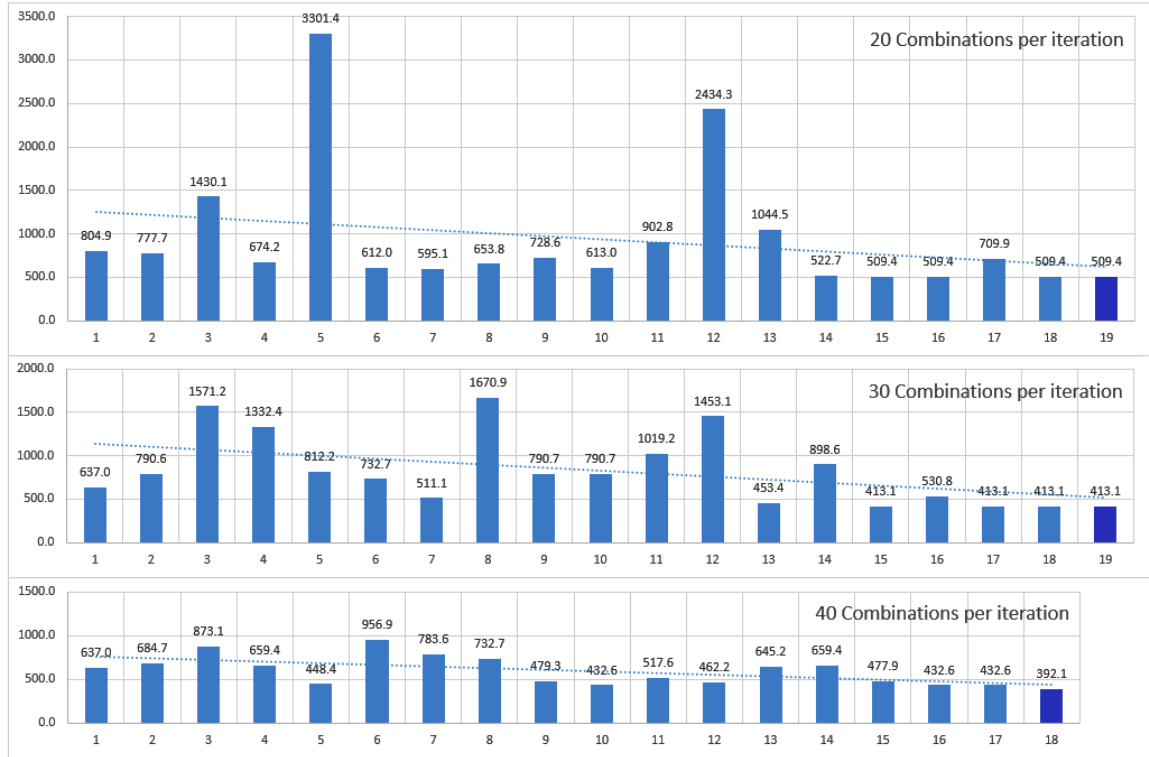


Figure 5.4: Heuristic solution convergence profile for increasing number of sampled combinations per iteration

Once the combinations with the least potential to be part of competitive solutions have been identified, we can prune at the end of each iteration one or more of the identified combinations from their respective sensors. This allows to progressively decrease the search space in each iteration. However, limited sampling implies that any combination considered to be pruned has a small potential to lead the solution search to a local minimum. Consequently, the more combinations are pruned per iteration, the more likely is to arrive at a local minimum solution. Thus, pruning only one combination per iteration carries the least chance of arriving at a local minimum solution. In turn, this involves more iterations and a correspondingly larger cumulative sample amount. In this

setting, one can consider pruning multiple combinations per iteration as a trade-off between lower solution search time (less number of iterations) and solution quality (gap to optimal value).

Sample count/ Iteration	20			30			40		
	Sensor	Remaining/Pruned combinations	Min. votes	Sensor	Remaining/Pruned combinations	Min. votes	Sensor	Remaining/Pruned combinations	Min. votes
1	d	5/[P, S]	0	d	5/[P, S]	0	d	5/[P, S]	0
2	d	4/[P, Q]	0	b	2/[Q, R]	0	b	2/[Q, R]	0
3	f	5/[P, R]	0	d	4/[P, Q]	0	g	5/[Q, S]	0
4	f	4/[Q, R]	0	f	5/[P, R]	0	f	5/[Q, R]	0
5	g	5/[P, S]	0	d	3/[Q, R]	0	a	2/[R]	0
6	g	4/[P, R]	0	a	2/[P]	1	d	4/[P, Q]	0
7	a	2/[Q]	0	f	4/[P, Q]	0	f	4/[P, R]	0
8	d	3/[P, R]	1	g	5/[Q, R]	0	g	4/[Q, R]	0
9	d	2/[R, S]	1	g	4/[Q, S]	0	f	3/[P, Q]	0
10	f	3/[Q, S]	1	d	2/[P, R]	1	d	3/[P, R]	1
11	f	2/[P, Q]	0	f	3/[Q, R]	1	g	3/[P, Q]	0
12	e	2/[R]	1	f	2/[Q, S]	1	e	2/[Q]	1
13	g	3/[Q, S]	0	e	2/[S]	1	d	2/[R, S]	1
14	a	1/[R]	2	g	3/[P, S]	0	e	1/[R]	2
15	b	2/[P, Q]	1	g	2/[R, S]	0	d	1/[Q, S]	1
16	g	2/[Q, R]	2	f	1/[P, S]	1	g	2/[P, S]	1
17	c	1/[P]	1	e	1/[R]	2	f	2/[Q, S]	2
18	e	1/[Q]	3	d	1/[Q, S]	3	N/A	N/A	N/A
Average min. votes per pruned comb.			0.72			0.61			0.53

Table 5.4: Pruned combinations over successive iterations

We illustrate the solution convergence profiles (Figure 5.4) and the corresponding pruned combinations over successive iterations (Table 5.4) for increasingly higher values of sample count (20, 30 and 40) per iteration. In each case, a single candidate combination is pruned per iteration. As shown for the sample count of 40, we note that since only one combination is pruned per iteration, the successively pruned combinations (after the first pruned combination), are different than the evaluated candidate combinations for the first iteration. This relates to the potential identification of more or less costly best solutions in successive sampling iterations, along with their respective elite solution groups and sensor assignments, based on which pruning candidates are determined. However, we can observe that for the sample count of 40, the successively pruned combinations, while determined over different elite group solutions, have no votes for the first 9 iterations and at most 1 or 2 votes for the subsequent iterations.

In Table 5.4, we can also note that the average minimum votes per pruned combination is decreasing when the sample count is increasing. This reflects that the pruned combinations are segregated with better confidence when the sample count is larger. Moreover, in Figure 5.4, we can see for each sample count value, the best solution values (rounded to the first decimal)

obtained over successive iterations along with their respective trend-lines. We can note that for a sample count of 20, there are quite notable differences between the encountered solutions, with a maximum at 3301.4 in iteration 5 and minimum at 509.4 in iterations 15, 16, 18 and 19. Then, for a sample count of 30, there are still somewhat notable differences with a maximum at 1670.9 in iteration 8 and minimum at 413.1 in iterations 15, 17, 18 and 19. Finally, for a sample count of 40, there are comparatively less notable differences with a maximum at 956.9 in iteration 6 and minimum at 392.1 in iteration 18. We can observe that for a sample count of 20 and 30, the solution search ends up in a local minimum at 509.4 and respectively at 413.1 after 19 iterations. In contrast, for a sample count of 40, the solution search ends up in the optimal value (rounded to first decimal) at 392.1 after 18 iterations.

We also note that for each sample count value, in the last iteration, the solution search space typically contains less elements than the sample count value. Hence, an exhaustive search can be performed at this stage with the same or less computational cost than sampling but with the guarantee of providing the best solution in the remaining search space. This is the reason for having 18 iterations for a sample count of 40 (rather than 19 as in the other two cases) since the remaining combination count in iteration 18 is under 40.

We discuss next the same case study problem in the context of using an available energy budget to limit the energy use of the sensor network. We consider without loss of generality that each sensor requires an amount of one energy unit to focus on a particular target. Moreover, we consider that sensors with the capacity of focusing on more than one target, require the corresponding amount of energy units matching their capacities. However, the availability of a limited energy budget may restrict some sensors from using all of their available target focusing capacity or even prevent some sensors from focusing on any target.

In order to better illustrate the effect of an energy budget in the context of the case study problem, we reconsider the problem in a setting where less precise target localization is allowed as trade-off for energy saving. Consequently, we allow for a minimum of 2 sensors per target as acceptable instead of 3. While this results in extra network capacity compared to the case where a minimum of 3 sensors are required per target, the solution search space of the problem actually increases since each sensor can partially use its capacity, thus increasing the number of possible sensor to target assignment combinations.

Having the requirement of minimum 2 sensors per target and a total of 4 targets corresponds to a minimum energy budget of 8 energy units. Furthermore, the summed target focusing capacities of the 7 sensors is given by $a(1) + b(2) + c(1) + d(2) + e(1) + f(2) + g(2)$, resulting in a maximum energy budget of 11 units. Thus, we solve the problem for an energy budget ranging from 8 to 11 units of energy.

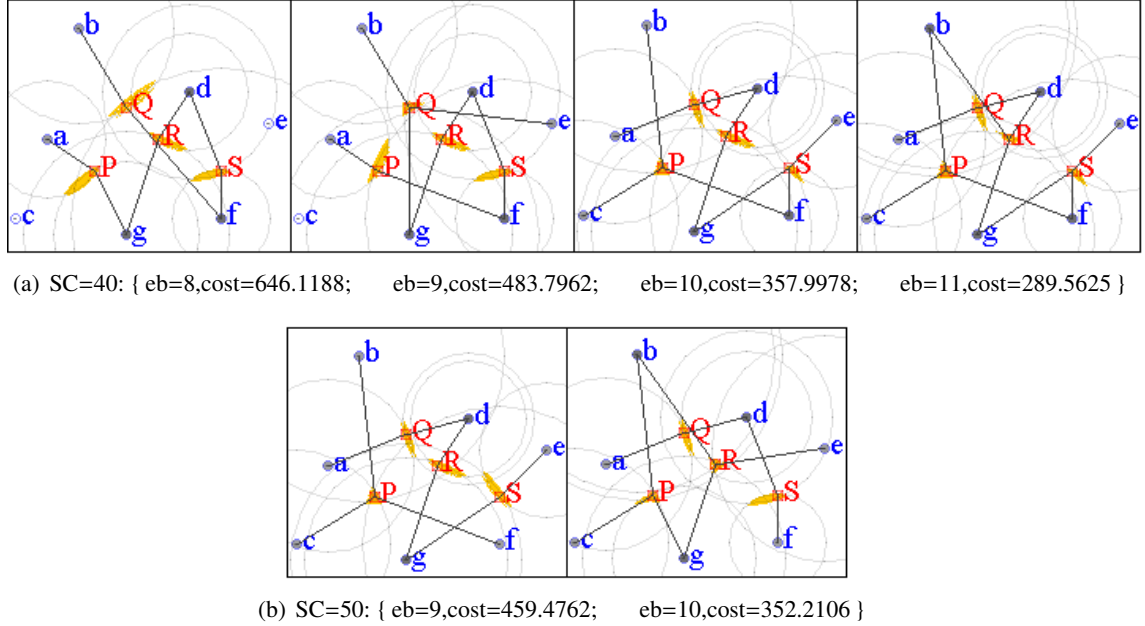


Figure 5.5: Solution cost comparison for increasing energy budget (8, 9, 10, 11)

Figure 5.5 presents, for increasing energy budget, the cost comparison between the solutions obtained with sampling count of 40 and the more cost effective ones obtained with sampling count of 50. Figure 5.5(a) depicts the solutions obtained for increasing energy budget values (8,9,10,11) using sample count of 40 (initially chosen as the solution search space is now larger, as previously mentioned). However, only for energy budget values of 8 and 11, the obtained solutions have been confirmed as optimal after exhaustive search. Increasing the sample count to 50, allows the generation of better solutions for energy budget values 9 and 10, shown in Figure 5.5(b), and confirmed as optimal after exhaustive search. We can see in each solution presented in Figure 5.5 the corresponding sensor-target assignment (lines connecting sensors to targets) and the related per target localization cost (area delimited by the range circle intersection of the assigned sensors).

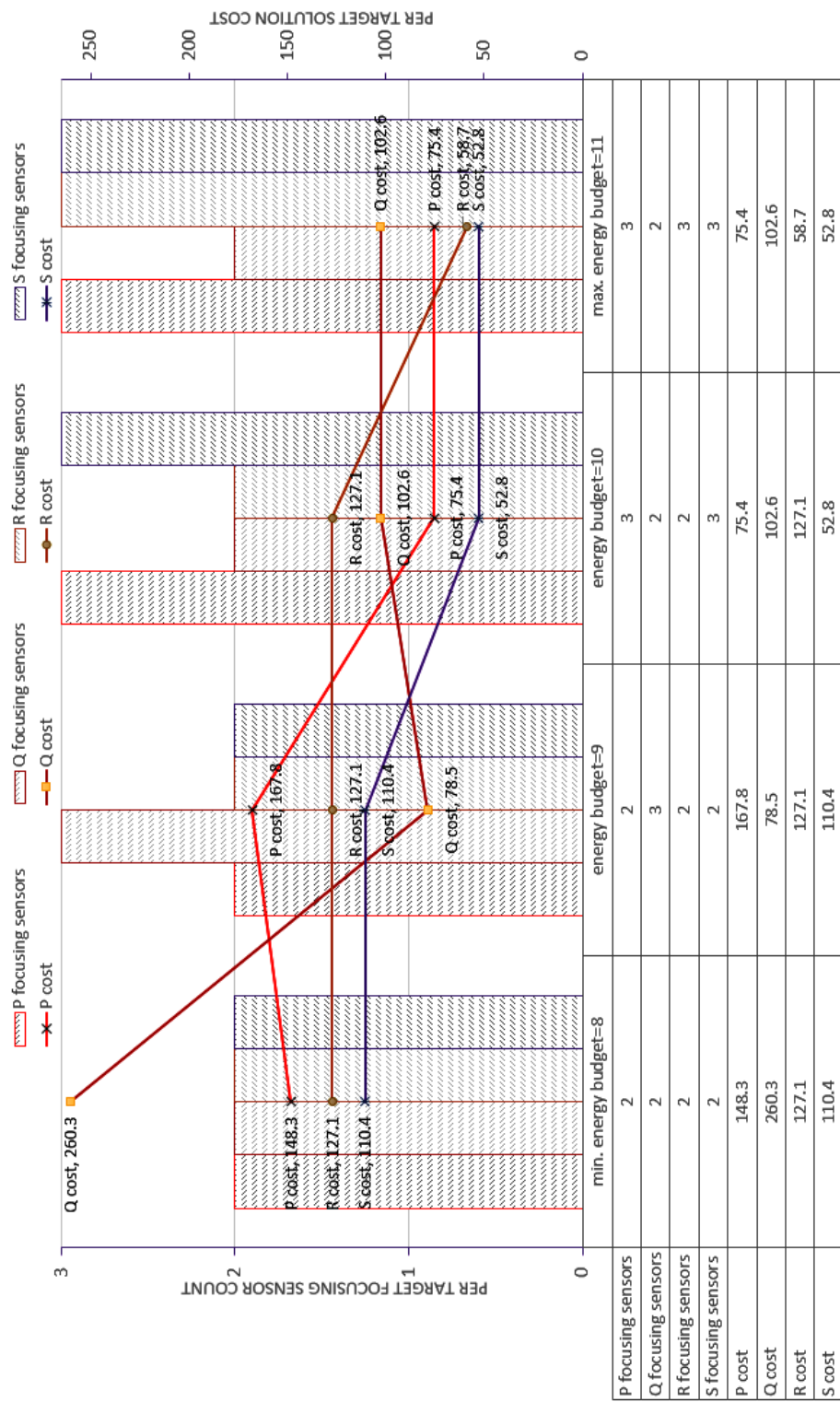


Figure 5.6: Solution characteristics for increasing energy budget relative to 'per target focusing sensor count' and 'per target solution cost'

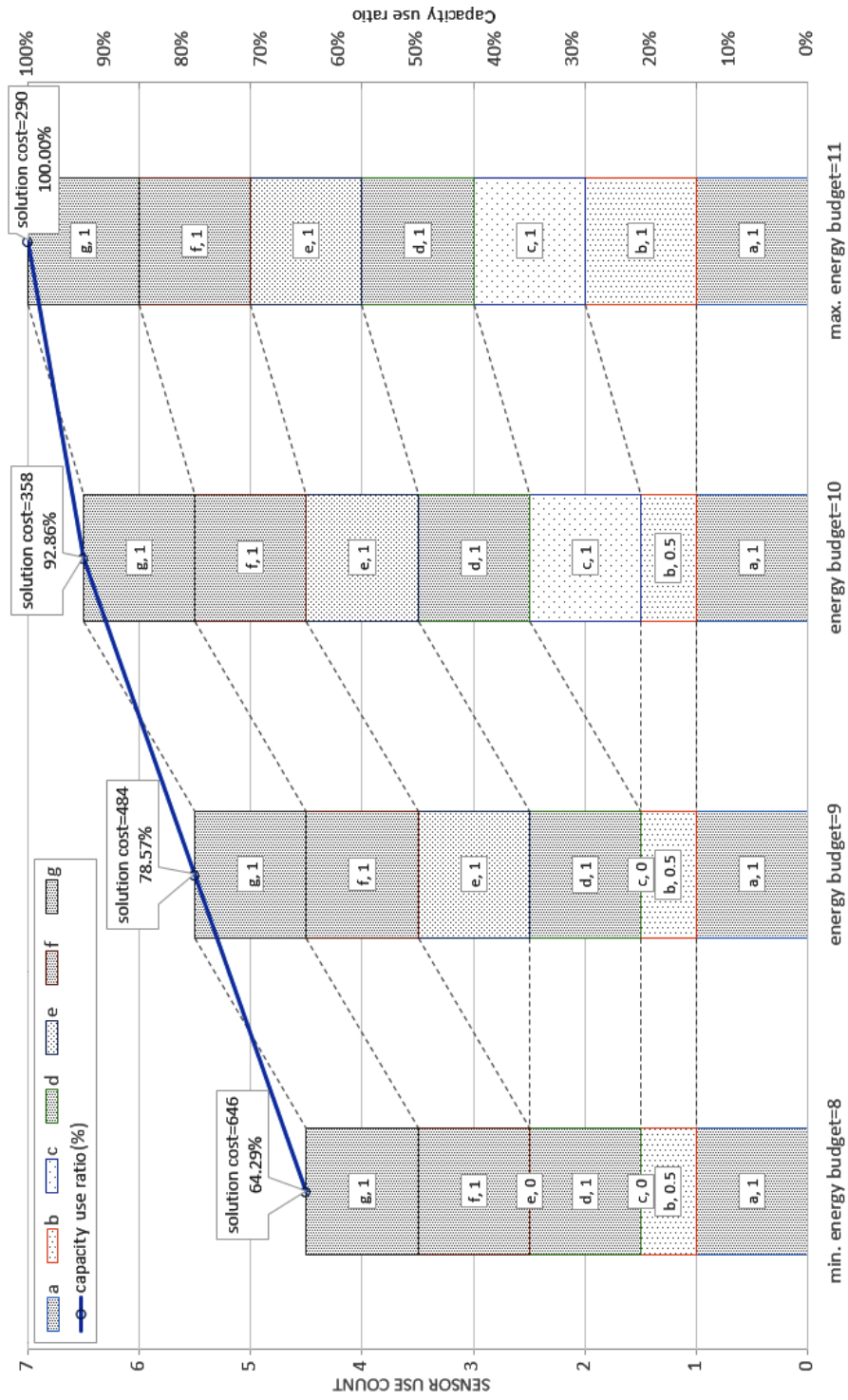


Figure 5.7: Solution characteristics for increasing energy budget relative to 'sensor use count' and 'network capacity use ratio'

As expected, increasing the energy budget leads to more cost effective solutions. We can also note in Figure 5.5(a) that for the smallest energy budget value of 8, two sensors (c and e) are not used while for energy budget value of 9, only one sensor (c) is not used.

Figure 5.6 depicts the characteristics of the obtained solutions (using sample count of 40), for increased energy budget in terms of the count of focusing sensors per target (target assignment count) as well as the solution cost breakdown per target. We note that the cost for each particular target does not always decrease with increased energy budget such as in the case of target Q, which has the highest cost for energy budget of 8 and the lowest cost for energy budget of 9. Also, target P has the highest cost for energy budget of 9 and the lowest cost for energy budget values of 10 and 11. Moreover, we note that increased values of energy budget allows the assignment of 3 sensors for an increasing number of targets, thereby lowering the cost. We can observe that for energy budget of 11 (which allows obtaining the best solution), the assignment of sensors to targets is the same as in the best solution of the original case study problem. However, the latter has a higher cost since it requires 3 sensors per target, thus incurring a penalty for one target, namely Q.

Figure 5.7 depicts, for increasing energy budget, the characteristics of the obtained solutions (using sample count of 40) in terms of the sensor capacity usage as well as the sensor network capacity use ratio (the corresponding solution cost is also shown as integer value for brevity). The smallest capacity use ratio is achieved for the minimum energy budget of 8 where sensor b is used at 50% capacity while sensors c and e are used at 0% capacity (not used). This corresponds to an overall sensor network capacity use of 64.29% given by the average capacity use ratio over all sensors: $(a(1)+b(0.5)+c(0)+d(1)+e(0)+f(1)+g(1))/7 = 4.5/7$ (4.5 sensors used out of 7). However, in this case, the localization cost is the highest (646). Bringing energy budget to 9 allows 100% use of sensor e, which results in a sensor network capacity use of 78.57% and an improved localization cost (484). Increasing energy budget to 10 also allows 100% use of sensor c, resulting in 92.86% network capacity use and better localization cost (358). Finally, for maximum energy budget, all sensors are used at 100% capacity with the best localization cost (290).

In Figure 5.8, we can see a comparison between the solutions obtained for increasing energy budget with sample count of 40 and respectively 50, relative to the sensor network capacity use and localization solution cost. We note that for the energy budget of 9, the contrast of the obtained solutions is the most pronounced as follows. For sample count of 40, the solution cost is 484 and the

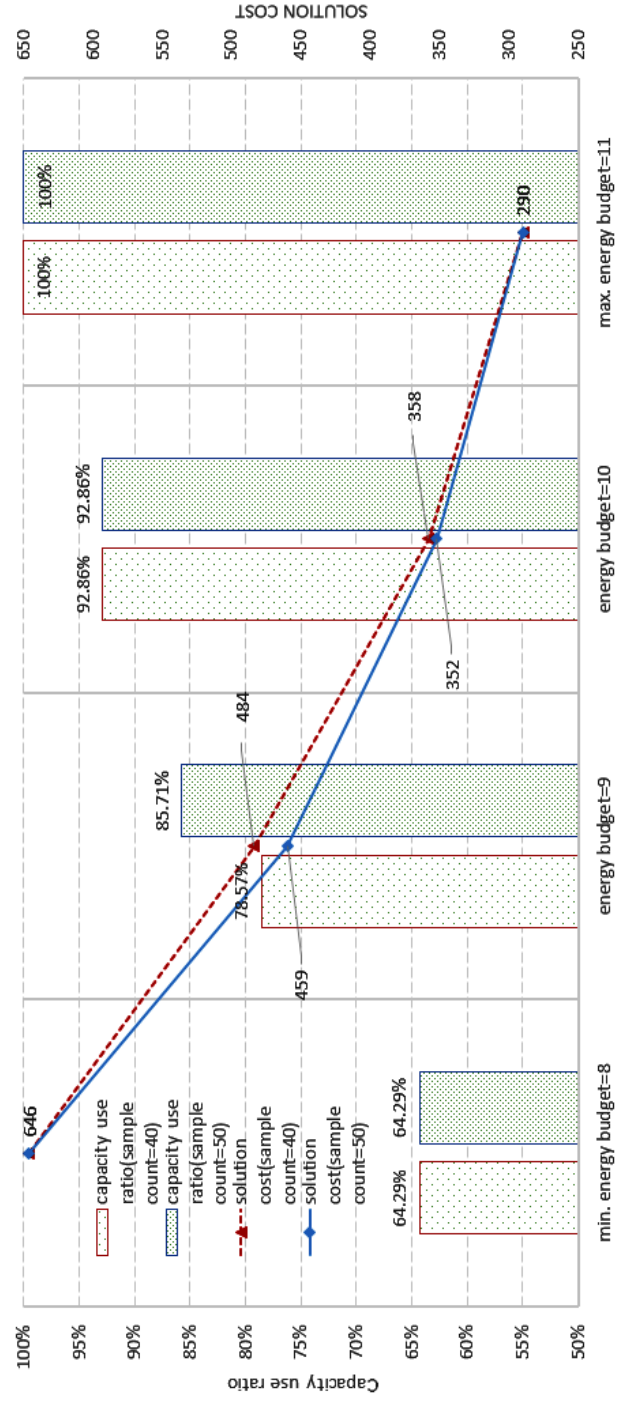


Figure 5.8: Solution comparison (sample cnt. 40/50) for increasing energy budget relative to 'network capacity use ratio' and 'solution cost'

network capacity use is 78.57% given by $(a(1)+b(0.5)+c(0)+d(1)+e(1)+f(1)+g(1))/7 = 5.5/7$. In contrast, the solution obtained with a sample count of 50 provides a better solution cost of 459 and a network capacity use given by $(a(1)+b(0.5)+c(1)+d(1)+e(1)+f(0.5)+g(1))/7 = 6/7$ (85.71%). This results from bringing sensor c to 100% use while reducing the use of sensor f to 50% capacity. The solution with cost of 484 spends the energy budget by focusing 2 sensors on targets P, R, S and 3 sensors on target Q whereas the solution with cost 459 spends the energy budget by focusing 2 sensors on targets Q, R, S and 3 sensors on target P .

5.5 Benchmark Results

In context of the present problem and associated model, we assessed the proposed approach experimentally using a problem set generated from known vehicle routing problem (VRP) instances [13]. The forenamed VRP data set was selected since it provides node arrangements relevant for supply chain activities and contains instances with sizes ranging from 20 to 101 nodes.

Problem	p-n20	p-n23	p-n40	p-n45	p-n50	p-n51	p-n55	p-n60	p-n65	p-n70	p-n76	p-n101
nodes	20	23	40	45	50	51	55	60	65	70	76	101
sensors	10	12	20	23	25	26	28	30	33	35	38	51
targets	10	11	20	22	25	25	27	30	32	35	38	50
sensors/targets ratio	1.000	1.091	1.000	1.045	1.000	1.040	1.037	1.000	1.031	1.000	1.000	1.020
min. sensor capacity	1	1	1	1	1	1	1	1	1	1	1	1
max. sensor capacity	4	4	4	4	5	5	5	5	6	6	6	6
avg. sensor capacity	3	2.75	3	2.87	3	2.88	2.89	3	2.9	3	3	2.94
min. target coverage	4	4	4	4	4	4	4	4	4	4	4	4
max. target coverage	7	7	7	7	10	9	8	9	12	8	10	11
avg. target coverage	4.7	5.2	4.5	5.4	5.6	5.7	5.7	5.6	6.2	5.7	5.9	6.5
solution search space order of magnitude >	10^8	10^9	10^{14}	10^{22}	10^{24}	10^{25}	10^{28}	10^{31}	10^{38}	10^{39}	10^{40}	10^{63}

Table 5.5: Benchmark problems

The aforementioned data set provides a more suitable context than using purely random generated instances as previously adopted in the context of related problems and their specific associated models. For each instance, half of the nodes have been considered as sensors and the other half as targets. For the instances with odd number of nodes, we selected one more sensor

compared to the number of targets. Only the selection of the sensor nodes was performed pseudo-randomly but in a manner whereby their capacity and range would make each problem challenging in the following sense. Each sensor has more targets in its range than its capacity and every target is covered by 4 or more sensors that have the target in their sensing range. The detailed breakdown for each of the benchmark problems is provided on a separate column in Table 5.5. In the latter, we can see that the sensor to target ratio is greater than unity for the odd node count instances since one more sensor node is available. The problems are divided in three subsets based on the maximum target focusing ability of the sensors, as follows. The problem set with 20, 23, 40 and 45 nodes have sensors with target focusing capacities from 1 to maximum 4 targets, with an average target coverage ranging from 4.5 to 5.4. The problem set with 50, 51, 55 and 60 nodes have sensors with target focusing capacities from 1 to maximum 5 targets, with an average target coverage ranging from 5.6 to 5.7. The problem set with 65, 70, 76 and 101 nodes have sensors with target focusing capacities from 1 to maximum 6 targets, with an average target coverage ranging from 5.7 to 6.5.

In order to make the problems challenging to solve without penalty, no reserve sensor capacity was considered. Thus, for all problems, we have an average sensor capacity of 3 or slightly lower than 3 for the odd node count instances (due to the presence on an additional sensor node). We can also note in Table 5.5, the progressively increasing solution search space order of magnitude with each larger problem instance, ranging from 10^8 combinations for the smallest instance of 20 nodes, up to 10^{63} combinations for the largest instance of 101 nodes.

5.5.1 Parameter Exploration and Performance Assessment

The performance assessment of 20, 23, 40 and 45 node problems with maximum sensor capacity of 4, is depicted in Figure 5.9 for different parameter values. Moreover, Figure 5.10 depicts for different parameter values, the performance assessment of 50, 51, 55 and 60 node problems where the maximum sensor capacity is 5. Furthermore, Figure 5.11 depicts for different parameter values, the performance assessment of 65, 70, 76 and 101 node problems where the maximum sensor capacity is 6. In each of the aforementioned three figures, we have 4 sub-figures, with the results for each of their respective problems. Each sub-figure depicts five *hs* and *hsmh* results, which represent average values obtained after running the heuristic with eight random seeds for different parameter values for the pruned combinations and sample count per iteration.

In the leftmost side of each sub-figure, we have one pruned combination and 1000 samples (1/1000). After that, we have two pruned combinations and 300 samples (2/300) followed by two pruned combinations and 100 samples (2/100). Then, we have three pruned combinations and 300 samples (3/300). Finally, in the rightmost side, we have three pruned combinations and 100 samples (3/100). For each of the parameter values, we can see two line graphs and two bar graphs. The line graphs depict the gap to the best solution results for *hs* (blue) and *hsmh* (gray), while their total number of combinations explored during solution search is depicted by the red and orange bar graphs respectively. The corresponding values are provided in the tables situated below the graphs. For each problem instance, the best solution to compare against was selected as the one with the smallest value among all experimental results. Since for a given problem, the evaluation of any particular sensor-target assignment is on the average equally costly in terms of computation, this can serve as an appraisal of the computing budget corresponding to particular parameter values. As reference for the performed experiments, the average number of sensor-target assignment combinations that can be evaluated on a Core i7 platform, using a Java implementation, is on the order of 1000000 per second.

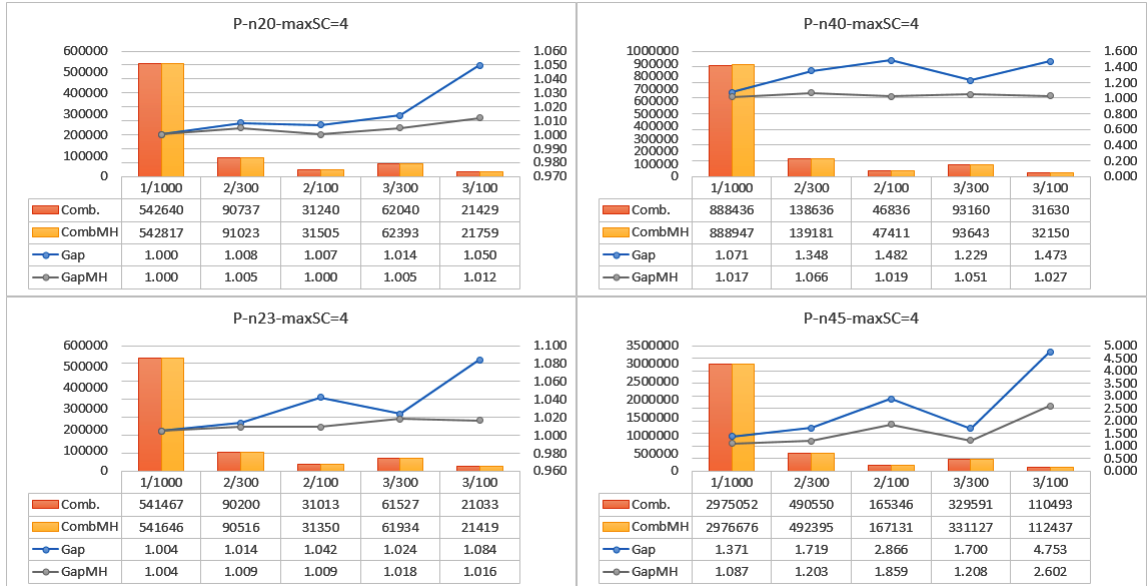


Figure 5.9: Performance assessment for the 20, 23, 40 and 45 node problems where the maximum sensor capacity is 4

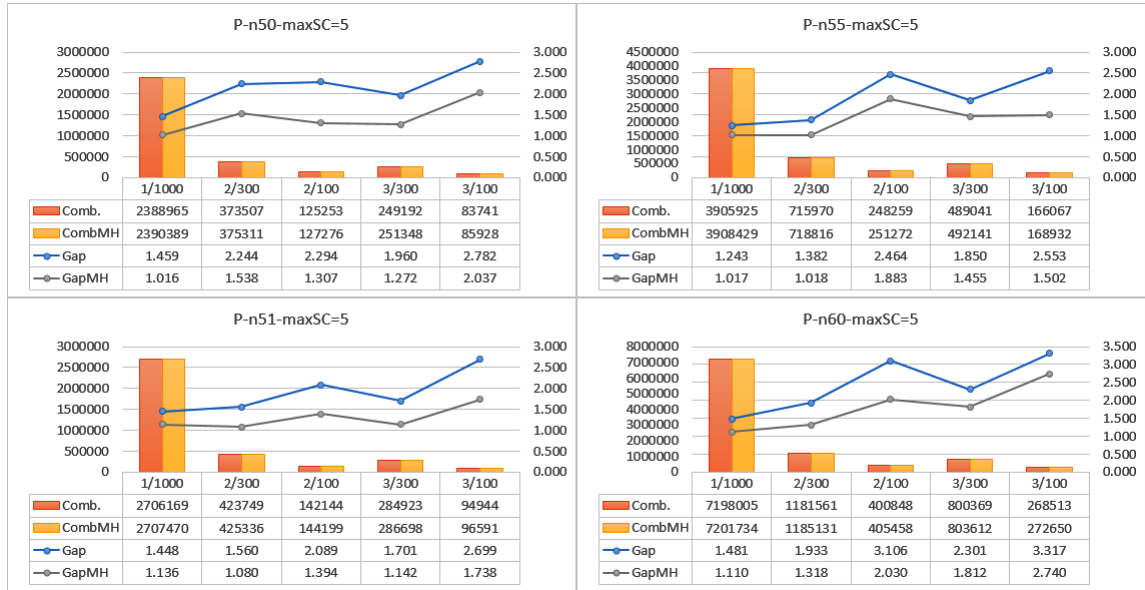


Figure 5.10: Performance assessment for the 50, 51, 55 and 60 node problems where the maximum sensor capacity is 5

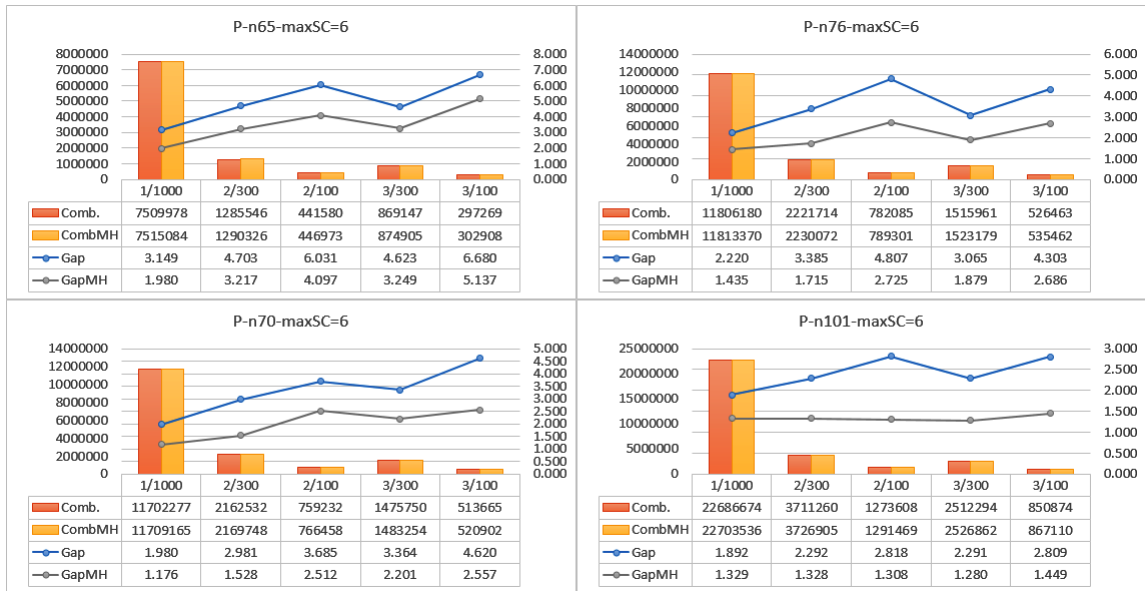


Figure 5.11: Performance assessment for the 65, 70, 76 and 101 node problems where the maximum sensor capacity is 6

We can see that the application of the meta-heuristic improvements slightly increases the total number of sensor-target assignment combinations. The selected parameter values have been chosen based on the following considerations. For 1/1000, we let the search procedure explore a large number of samples while pruning only one combination per iteration. This allows to obtain the best results but the total number of explored combinations is large, which corresponds to large execution time. For 2/300, the search procedure is notably faster but the results are not as good compared to 1/1000. However, we can note that the meta-heuristic improvements have notable benefits on the results at the expense of only minor increase in the explored sensor-target assignment combinations, due to the application of the meta-heuristic. For 2/100, the search procedure is even faster but the results are degraded and the application of the meta-heuristic improvements is not as effective. For 3/300, the results are somewhat less competitive in general compared to 2/300, hinting that pruning an additional combination per iteration allows for faster result generation at the expense of slightly more costly results. Finally, for 3/100, the search procedure is fastest but the results are the least competitive while the meta-heuristic improvements have limited benefits due to insufficient sampling and more aggressive pruning of sensor-target combinations.

5.5.2 Result Analysis

Table 5.6 shows the experimentally obtained benchmark results (integer rounded) presented in a separate column for each problem instance. The experiments have been conducted using additive penalty ($\rho_1 = 5000$) for the cases where less than 3 sensors are assigned to a target. We mention that in the table, the penalty values are displayed in multiples of 1000. For restricted energy budget, we used $\rho_2 = 6$. However any solution that exceeds the energy budget restriction can be adjusted to one respecting it, by deassigning the cost-wise least impacting targets from the sensor(s) that use(s) the most capacity.

The table is divided into three sections presenting the results obtained using full energy budget (100%), three quarters budget (75%) and half budget (50%) respectively. The first section contrasts four types of results, as follows: nearest-neighbour (*nn*) solutions, nearest-neighbour solutions followed by meta-heuristic improvements (*nnmh*), heuristic solutions (*hs*) and heuristic solutions followed by meta-heuristic improvements (*hsmh*). The meta-heuristic solution improvements are carried out after restoring the initial allocation choices for all the sensors followed by

Problem	p-n20	p-n23	p-n40	p-n45	p-n50	p-n51	p-n55	p-n60	p-n65	p-n70	p-n76	p-n101
With maximum budget ($K \times \sum_{i=1}^n C_i$), full (100%) energy budget - all sensors used at full capacity												
nn [no penalty]	2791	2880	8061	5061	66407	64109	6726	7310	74289	13211	14054	99381
nn penalty $\times 1000$	30	30	35	40	40	40	45	70	80	75	105	95
Tgts [0 sensors]	0	0	0	0	1	1	0	0	1	0	0	0
Tgts [1-2 sensors]	4	5	5	8	7	7	7	10	13	10	12	15
Tgts [3 sensors]	1	2	9	8	10	10	12	7	7	14	8	18
Tgts [4+ sensors]	5	4	6	6	7	7	8	13	11	11	18	17
nnmh [no penalty]	1371	4267	3433	7362	5519	3376	2903	4048	5952	4937	4519	86518
nnmh penalty $\times 1000$	0	10	10	20	20	35	25	35	35	40	35	80
Tgts [1-2 sensors]	0	1	2	4	4	6	5	5	5	7	5	13
Tgts [3 sensors]	10	8	16	15	17	12	17	18	22	20	26	22
Tgts [4+ sensors]	0	2	2	3	4	7	5	7	5	8	7	15
hs [no penalty]	1124	2181	1575	1024	5002	2538	1631	3042	4584	3131	3798	22869
hs penalty $\times 1000$	0	0	0	0	0	0	0	0	0	0	0	0
Tgts [3 sensors]	10	11	20	22	25	25	27	30	32	35	38	50
hsmh [no penalty]	1124	2181	1529	938	3794	1900	1438	2513	2519	1799	2194	14632
hsmh penalty $\times 1000$	0	0	0	0	0	0	0	0	0	0	0	0
Tgts [3 sensors]	10	11	20	22	25	25	27	30	32	35	38	50
With 75% of the energy budget												
hsmh [no penalty]	1172	2283	1725	1171	3068	2029	1620	2274	2094	2072	2229	21721
hsmh penalty $\times 1000$	40	45	75	85	95	95	105	115	120	135	145	190
Tgts [1 sensors]	0	0	0	0	1	0	1	1	0	0	0	0
Tgts [2 sensors]	8	9	15	17	17	19	19	21	24	27	29	38
Tgts [3 sensors]	2	2	5	5	7	6	7	8	8	8	9	12
Capacity use ratio	76%	76%	78%	75%	76%	80%	76%	77%	76%	71%	79%	76%
Sensor use	10/10	12/12	20/20	23/23	23/25	26/26	27/28	28/30	33/33	31/35	37/38	49/51
With 50% of the energy budget												
hsmh [no penalty]	5104	4730	14514	15105	16294	14738	14159	17235	16941	21829	16805	114216
hsmh penalty $\times 1000$	75	85	150	165	190	190	205	225	240	265	285	375
Tgts [1 sensors]	5	6	10	11	13	13	14	15	16	18	19	25
Tgts [2 sensors]	5	5	10	11	12	12	13	15	16	17	19	25
Capacity use ratio	59%	52%	51%	48%	53%	57%	55%	51%	53%	55%	58%	56%
Sensor use	10/10	10/12	15/20	19/23	23/25	26/26	28/28	30/30	33/33	35/35	38/38	50/51

Table 5.6: Benchmark results for asset localization

an iterative attempt at target swapping among sensor pairs covering the same targets in pursuit of lowering the overall cost. First, we discuss the results obtained by using the myopic *nn* sensor assignment. We can see that such assignment is very unfavourable since there are instances where certain targets have no (0) sensors assigned. There is also a large number of targets that have either under-assigned (1-2 sensors) or over-assigned (4+) sensors while the number of targets with 3 assigned sensors is comparatively less. Next, we discuss the effect of subjecting the *nn* solutions to the meta-heuristic improvements. We can note that the *nnmh* solutions no longer contain targets with no sensors assigned. Also, the number of targets that have either under-assigned or over-assigned sensors is lower compared to the number of targets with 3 assigned sensors. We can note that for the 20 node instance in particular, all targets have 3 sensors assigned. Thus, the obtained

nnmh solution results indicate the usefulness of the meta-heuristic improvements. However, the scope of improvement over the *nn* solutions is limited by their initial unfavourable sensor-target assignments. This becomes evident when we inspect the *hs* solutions, which are notably better compared to the *nnmh* solutions. Thus, all *hs* solutions have 3 sensors assigned for each target, thus incurring no penalties. We can see that even for the 20 node instance, where the *nnmh* solution has no penalty, the *hs* solution still has a better cost. This hints that the heuristic search is not only able to avoid the penalties but the obtained assignment is quite cost effective.

With respect to the *hsmh* solutions obtained after subjecting the *hs* solutions to the meta-heuristic improvements, we can note that they exhibit for the most part improved costs, especially for the larger instances. Also, the *hsmh* solution values for the 20, 23 and 40 node instances are optimal for the first two (10^8 and 10^9 order of magnitude) and most likely optimal for the third one (10^{14} order of magnitude). This was confirmed via exhaustive search for the first two instances. The third instance was assessed by removing certain combinations (with no votes across millions of sampled cost effective solutions) followed by an exhaustive search of the remaining (10^{12} order of magnitude) space. We also note that for the 20 and 23 node instances, the *hs* solutions obtained are also optimal. Thus, the meta-heuristic improvements offer most benefits when the solution search space order of magnitude is larger. The second and third sections of Table 5.6, present the results obtained for the same problem instances when the energy budget is restricted. Since the benchmark problems are tight (no reserve capacity), the restriction of the energy budget involves the application of penalties, which will be discussed next.

5.5.3 Impact of Energy Budget Restriction

The restriction of the energy budget results in solutions where certain sensors are unused or underused (used at less than full capacity). Such restriction allows to save energy with the benefit of allowing the network to operate longer but depending on the degree to which the energy budget is restricted, the localization cost may increase. In addition, it may no longer be possible to respect the requirement to have a certain number of sensors assigned per target, which corresponds to solutions exhibiting penalties. Also, for larger problem instances, separate penalty may be incurred at the heuristic search stage in case that the solution exceeds the energy budget restriction. However, such penalty can be mitigated by removing a number of sensor-target assignments (from

the sensors with the highest capacity use) whereby the solution cost is allowed to increase such that the energy restriction can be satisfied. Furthermore, this procedure is applied, if needed, during the meta-heuristics such that any remaining penalty is resulting from the assignment of less than 3 sensors per target.

In Table 5.6, second section (corresponding to 75% energy budget), we observe that all solutions exhibit penalties since the sensor network cannot use its full capacity. As such, the sensor network capacity use ratio varies from 71% (p-n70) to 80% (p-n51). Regarding the degree to which the sensors are used, we can notice that some sensors are unused in certain cases, ranging from 1 (p-n55, p-n76) to 4 (p-n70). Also, we note that for most problems, the solutions predominantly assign 2 sensors for most targets while the number of targets with 3 sensors assigned is notably smaller. There are also a few problems (p-n50, p-n55 and p-n60) where one target is assigned only one sensor. If we compare the solutions by disregarding the penalties, we can see that in most cases the solution values have higher cost compared to those obtained in the case where 100% energy budget is used. There are a few exceptions (p-n50, p-n60 and p-n65) but in such solutions, while some targets can be localized relatively well even with 2 sensors (given the particular node arrangements of those problems), a few targets are very poorly localized, especially when being assigned only one sensor for some targets. Such exceptions stem from the use of assignment choices typically unavailable when we can assign 3 sensors per target in order to avoid penalties.

In the third section of Table 5.6 (50% energy budget), we can observe that all solutions exhibit significant penalties (the sensor network is severely restricted). The sensor network capacity use ratio varies from 48% (p-n45) to 59% (p-n20). We also note that for all problems, the obtained solutions assign only 1 or 2 sensors per target. Even if we disregard the penalty, the corresponding solutions have significantly higher cost even when compared to those obtained in the case where 75% energy budget is used. With respect to sensor use, some sensors are also unused in certain cases, ranging from 1 (p-n101) to 5 (p-n40). For all three energy budget levels (100%, 75% and 50%), Figure 5.12 shows the average sensor capacity use corresponding to the different energy budgets used to generate the solutions for each of the benchmark problems. Figure 5.12 also shows the minimum and maximum sensor capacity for each problem.

We discuss next the effect of energy budget restriction on the solution obtained for the first problem (p-n20) from the benchmark results presented in Table 5.6. The problem has 10 sensors

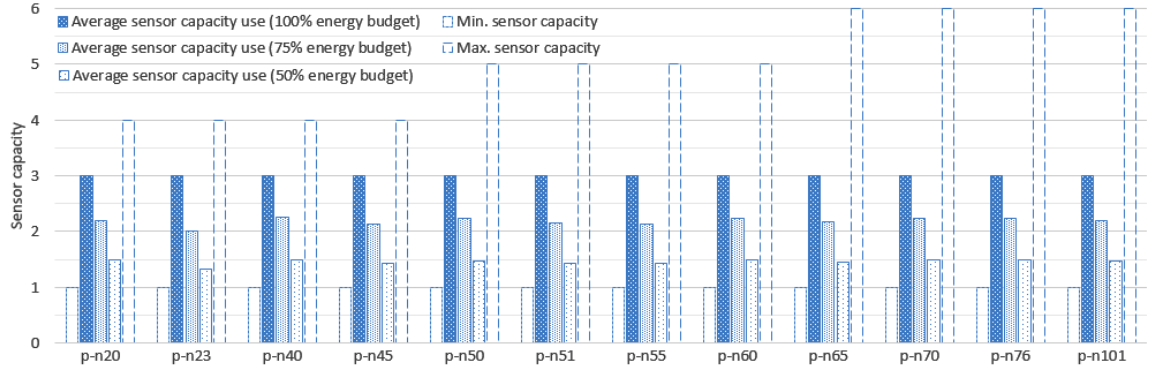


Figure 5.12: Average sensor capacity use relative to the benchmark results obtained for each problem instance with different energy budgets

.	Sensor	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
	Range	55	45	55	55	50	40	50	40	50	50
Target	Location	107,125	85,70	76,123	186,125	168,104	152,57	131,61	122,102	183,74	171,85
T1	152,72					(-)	(-)	(x)		(x)	(x)
T2	143,87	(-)				(-)	(x)	(-)	(x)	(-)	(x)
T3	152,116	(x)			(-)	(x)			(-)		(x)
T4	107,93	(-)	(-)	(x)				(x)	(x)		
T5	122,72		(-)				(x)	(x)	(x)		
T6	183,114				(x)	(x)				(x)	(-)
T7	171,45						(x)	(-)		(x)	(x)
T8	88,112	(x)	(x)	(x)					(-)		
T9	180,57					(-)	(x)	(x)		(x)	(-)
T10	125,121	(x)		(-)		(x)			(x)		

Table 5.7: P-n20 (100% energy budget)

Target\	Sensor	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
	TAC \ CU	3/3=1	1/1=1	1/2=0.5	1/1=1	2/3=0.66	2/4=0.5	3/4=0.75	4/4=1	2/4=0.5	3/4=0.75
T1	2					(-)	(-)	(x)		(-)	(x)
T2	2	(-)				(-)	(-)	(-)	(x)	(-)	(x)
T3	2	(x)			(-)	(x)			(-)		(-)
T4	3	(-)	(x)	(-)				(x)	(x)		
T5	2		(-)				(-)	(x)	(x)		
T6	2				(x)	(-)				(-)	(x)
T7	2						(x)	(-)		(x)	(-)
T8	2	(x)	(-)	(x)					(-)		
T9	2					(-)	(x)	(-)		(x)	(-)
T10	3	(x)		(-)		(x)			(x)		

Table 5.8: P-n20 (75% energy budget)

with an average capacity of 3 and 10 targets such that the full energy budget (needed to assign 3 sensors for each target) has a value of 30. Table 5.7 presents the problem data (sensor and target locations as well as target coverage) with the sensor-target assignment solution for full energy budget where 3 sensors are assigned for each target (target assignment count = 3) and each sensor is used at full capacity (capacity use = 100%).

Target\	Sensor	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
	TAC \ CU	2/3=0.66	1/1=1	1/2=0.5	1/1=1	3/3=1	1/4=0.25	3/4=0.75	1/4=0.25	1/4=0.25	1/4=0.25
T1	2					(-)	(-)	(x)		(-)	(x)
T2	2	(-)				(x)	(-)	(x)	(-)	(-)	(-)
T3	2	(x)			(-)	(x)			(-)		(-)
T4	2	(-)	(x)	(-)				(-)	(x)		
T5	1		(-)				(-)	(x)	(-)		
T6	1				(x)	(-)				(-)	(-)
T7	1						(x)	(-)		(-)	(-)
T8	1	(-)	(-)	(x)					(-)		
T9	1					(-)	(-)	(-)		(x)	(-)
T10	2	(x)		(-)		(x)			(-)		

Table 5.9: P-n20 (50% energy budget)

Table 5.8 and Table 5.9 provide the sensor-target assignment solutions for restricted energy budgets corresponding to three quarters (75%) and respectively half (50%) of the maximum energy budget. Instead of locations, these two tables provide the target assignment count (TAC) and capacity use (CU) values for each target and sensor respectively.

In all three tables, each column provides the related information for each sensor while each row provides the details for each target. Each target row has markings in round brackets indicating the sensors covering the target. In this setting, simple sensor-target coverage is marked by (-) while assignment to a covered target is marked by (x). The corresponding benchmark solution cost is 1124(+0 penalty) for 100% energy budget, 1172(+40000 penalty) for 75% energy budget and 5104(+75000 penalty) for 50% energy budget.

We note that the sensor network is using progressively less capacity of its sensors with decreased energy budget according to the energy budget limitation. Thus, in Table 5.7, the use of full energy budget corresponds to a sum of target assignments of 30 while the network capacity use ratio is 100%. Then, in Table 5.8, for 75% energy budget, the target assignment sum decreases to $2 + 2 + 2 + 3 + 2 + 2 + 2 + 2 + 2 + 3 = 22$ (the limit value since $30 \times 0.75 = 22.5$) while the network capacity use ratio decreases to $(1 + 1 + 0.5 + 1 + 0.66 + 0.5 + 0.75 + 1 + 0.5 + 0.75)/10 = 0.766$ ($\approx 76\%$). Finally, in Table 5.9, for 50% energy budget, the target assignment sum decreases to $2 + 2 + 2 + 2 + 1 + 1 + 1 + 1 + 1 + 2 = 15$ while the network capacity use ratio decreases to $(0.66 + 1 + 0.5 + 1 + 1 + 0.25 + 0.75 + 0.25 + 0.25 + 0.25)/10 = 0.591$ ($\approx 59\%$).

Appendix A contains an extensive amount of additional tables providing the data for the benchmark problems with node counts ranging from 23 to 101 in terms of sensor and target locations as well as the best sensor-target assignment solutions for full energy budget.

5.6 Summary

This chapter proposed an efficient management approach for resource constrained WSN that allows to minimize overall localization error cost for the covered target assets. The contribution involves an evolutionary learning heuristic technique suitable for asset localization. The key benefit consist in providing an efficient assignment of multiple sensors, with different limitations, in terms of sensing ranges and focusing capacities, to multiple target assets. Moreover, the proposed technique also allows to effectively manage the WSN in the context of an available energy budget that restricts the full use of the sensors capacities in order to conserve energy. Employing the energy budget restriction as a part of WSN management for asset localization provides a corresponding trade-off between energy conservation and localization performance. Furthermore, an illustrative case study was presented both in absence and in the presence of energy budget restrictions.

In contrast to [147], which considers multi-sensor to multi-target tracking by finding the best match of sensor measurements to different targets, the approach in this chapter aims at effectively assigning sensors with different capacities to multiple targets. This allows localization error minimization while observing an energy budget restriction on sensor capacity use. With respect to [26], which aims at asset tracking via mobile devices, with energy use versus localization accuracy trade-off, the approach in this chapter considers sensors' energy use. Thus, sensors can underutilize their capacities to save energy as trade-off for accuracy. Compared to [60], which considers the assignment of two homogeneous sensors per individual target to minimize the overall expected target location error, the approach in this chapter considers variable assignment of heterogeneous sensors to multiple targets depending on the available sensor capacity and usable energy budget.

In terms of limitations, since the approach involves a learning-based technique, a typical challenge relates to the performance of solution sampling over the vast combinatorial solution search spaces of larger problems. In this case, the knowledge gathered by evolutionary learning can exhibit a limited effectiveness. Notwithstanding, the proposed technique is quite relevant for problems of practical sizes. In this respect, the proposed technique was assessed in various parameter settings via extensive benchmarks over a data set containing problems ranging from 20 to 101 nodes. The obtained results indicate the effectiveness of the proposed approach, which allows for an user applicable trade-off in terms of solution quality versus computation time.

Chapter 6

Logistic Planning with Risk Mitigation and Plan Adaptation

In hostile environments, the logistics distribution component of a plan can be highly impacted by exogenous events leading to route cost increase, potential changes in demand levels, vehicle failure, etc. In regard to vehicle failure, it also impacts the demand nodes (e.g., customers) that will not receive their expected deliveries.

In this setting, an important aspect for logistic support consists in having a planning/replanning system that can take into account logistic delivery risk mitigation and related plan adaptation requirements. This mandates fast solution generation techniques that can provide decision makers with appropriate capabilities of responding to changing circumstances during plan execution. In this pursuit, this chapter proposes a heuristic solution technique that extends the approach introduced in [132], to handle the underlying risk constrained multi-depot logistic planning problem along with replanning (adaptive planning) features. This involves the simultaneous mitigation of potential vehicle breakdown/loss and commodity delivery failure due to on-route risk exposure.

The considered problem extends the multi-depot vehicle routing problem (MDVRP) [66] where multiple vehicles from multiple depots have to make deliveries to a set of customer nodes with minimized travel cost. The extension considers the risk of vehicle failure and consequently the potential loss of vehicles from depots' side along with the potential loss of commodity from the customers' side. The problem relates to some extent to the risk-constrained cash-in-transit

vehicle routing problem [140], which augments the classical single depot vehicle routing problem [142] with a risk constraint that limits the risk of each vehicle.

Given that route planning and replanning are critical aspects for collaborative planning [131] and plan execution, the proposed approach can provide notable benefits for collaborative logistics support plan adaptation. As such, the heuristic allows for online near-optimal replanning (adaptive planning) with risk mitigation, in short or near-real time, while addressing potentially changing deliveries from multiple depots, where each depot may have a distinct decision maker.

6.1 Risk Constrained Multi-Depot Vehicle Routing Problem

In the following, we present an overview of the underlying research problem along with the considered assumptions and the risk on route evaluation technique before capturing the problem through a mathematical model.

6.1.1 Overview of the Problem

The Risk Constrained Multi-Depot Vehicle Routing Problem (RCMDVRP) addresses commodity delivery by a vehicle fleet (spread over multiple depot nodes) to demand nodes (also termed as customers) across a risk-prone transport network. The latter is captured by a user defined complete graph $G = (V, E)$ where V is the set of nodes and E is the set of edges specifying a quantitative distance relation on $(V \times V)$, typically derived from locations specified for the nodes. For any pair of nodes $\langle i, j \rangle \in (V \times V)$, based on their distance relation, we have a corresponding vehicle routing cost c_{ij} . Alongside, a user defined on-route vehicle failure risk probability ρ_{ij} is specified for vehicle leg traversal between nodes i and j . The fleet of vehicles is captured by set K and each vehicle $k \in K$ has a fixed cargo capacity Q_k . Moreover, an employed vehicle k departing from a particular depot, is bound to complete its tour by returning to the same depot. Furthermore, each vehicle k can be exposed to no more than a maximum tour risk threshold $maxRisk$ of vehicle failure.

The transport network includes both depots and customers. Each customer node i has a user defined demand d_i (integer) for commodity. Also, a user defined per unit demand (PD) loss cost is specified whereby for every demand d_i , we have a corresponding potential cargo loss cost in case of failed commodity delivery, depending on the cumulative on-route risk probability for the

vehicle serving node i . In contrast, depot nodes have no demand while hosting vehicles $k \in K$. Similarly, a user defined per unit vehicle (PV) loss cost is specified whereby for every employed vehicle, we have a corresponding potential vehicle loss cost in case of vehicle failure, depending on the cumulative tour risk probability of the vehicle returning to its departing depot.

In case of predefined depots, a solution for a RCMDVRP instance provides the routes for each employed vehicle to serve all demands such that the overall routing cost, overall expected vehicle loss cost and overall expected cargo loss cost is minimized. In the proposed formulation, we also consider the situation where the depots are not predefined, instead each node $i \in V$ is representing a potential candidate to be selected as depot. In this setting, a user defined depot establishment cost ec_i is specified for each node i . Consequently, the RCMDVRP solution minimizes in this case the combined routing cost, expected vehicle and cargo loss costs, along with the overall depot establishment cost for the selected depots.

We note that the problem turns into classical MDVRP if the risk across every edge of the transport network graph is 0 (the potential vehicle and cargo loss cost is 0 and any vehicle tour has 0 risk). Since MDVRP has NP-hard complexity [37], it follows that RCMDVRP is also NP-hard since MDVRP is a particular case of RCMDVRP where no risk is considered in the transport network. Thus, exact analytical solution generation for RCMDVRP is computationally prohibitive in general. In turn, heuristic methods typically provide near-optimal solutions (in bounded computation time and memory space) and even optimal solutions for certain small size problems.

6.1.2 Assumptions

The problem is considered under the following assumptions. Every customer is served by only one vehicle (there is no split demand serving) and for each customer, at least one vehicle can serve its demand. The overall cargo capacity of all vehicles is large enough to serve all demands (the availability of more vehicles allows for potential vehicle losses while still allowing to serve all demands in case of replanning). A vehicle directly visiting any customer and then returning to its depot is exposed to no more than the maximum threshold of vehicle tour risk. When depots are not predefined, a parameter sets the maximum number of potential depots. In this setting, any selected depot node serves its own demands and the depot establishment cost values are considered as large enough to preclude a trivial solution where most or all nodes are selected as depots.

6.1.3 Risk on Route Evaluation Technique

We present in the following the employed concept of risk on route evaluation. For clarity of explanation, we assume that the consequence of a risk-related event results in vehicle failure. Therefore, the impact of such an event (i.e., a vehicle loss) also corresponds to cargo loss since the vehicle will not be able to deliver its payload to rest of the customers on the remainder of its route.

Figure 6.1 depicts the impact of risk on a vehicle tour through a simple problem. As per the illustration, the problem (left side of the figure) has 4 nodes, which include 3 customers, namely A, B and C and 1 depot, namely D. The commodity demands of the customers (shown in round parenthesis) and are as follows: A(1), B(2) and C(3). Depot D has 1 vehicle with a capacity of 6 cargo units of commodity. Each edge presents a pair of values: the routing cost (3 for AC, CA, BD, DB; 2 for all other edges) and the traversal risk probability (0.1 for every edge). For more clarity, the values on each edge are bidirectional. In the depicted configuration, there exist 6 possible tours for the vehicle to serve the customers while starting and ending at depot D (middle of the figure). There are two vehicle tour routes with the best routing cost ($rCost$), namely DABCD ($rCost = 2 + 2 + 2 + 2 = 8$) and DCBAD ($rCost = 2 + 2 + 2 + 2 = 8$). The vehicle and cargo risk over these two routes is also illustrated via corresponding decision trees (right side of the figure).

Let us examine the potential vehicle and cargo loss due to risk exposure on route DABCD.

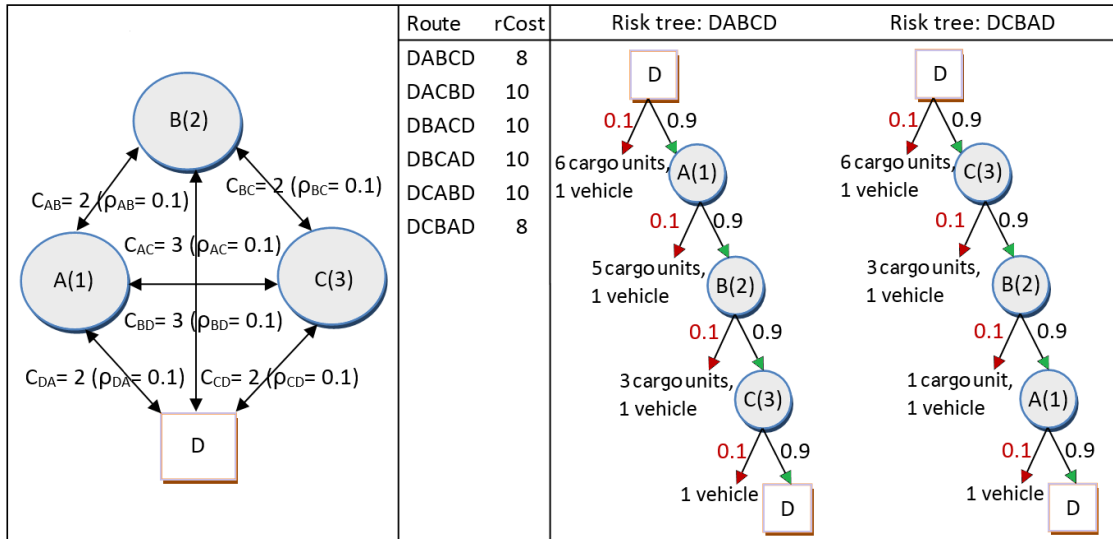


Figure 6.1: Risk on route example

If the vehicle fails at the beginning, the total loss is full 6 cargo units (and the vehicle). If the vehicle reaches A, then in case of a vehicle failure event on edge AB, the loss is 5 cargo units and 1 vehicle, since the vehicle served 1 cargo unit at A. If the vehicle reaches B, then in case of a vehicle failure event on edge BC, the loss is 3 cargo units and 1 vehicle, since the vehicle served 1 cargo unit at A and 2 cargo units at B. If the vehicle reaches C, then in case of a vehicle failure event on edge CD, there is no loss of cargo commodity, just the loss of 1 vehicle, since the vehicle served 1 cargo unit at A, 2 cargo units at B, and 3 cargo units at C. Following the decision tree, the risk of vehicle failure (rvf) on the routing path is obtained by subtracting from unity the overall chance of success given by multiplying each route segment success probability (1 - risk over the route segment):

$$rvf(DABCD) = 1 - (1 - 0.1) \times (1 - 0.1) \times (1 - 0.1) \times (1 - 0.1) = 1 - 0.6561 = 0.3439$$

Likewise, the risk of failing to deliver commodity for a demand node on this path is given by the risk of vehicle failure to reach that demand node:

- $A : rvf(DA) = 1 - (1 - 0.1) = (1 - 0.9) = 0.1$
- $B : rvf(DAB) = 1 - (1 - 0.1) \times (1 - 0.1) = (1 - 0.81) = 0.19$
- $C : rvf(DABC) = 1 - (1 - 0.1) \times (1 - 0.1) \times (1 - 0.1) = (1 - 0.729) = 0.271$

In this setting, for a given cost of PV per unit vehicle, the expected vehicle loss on route is:

$$\begin{aligned}
 DABCD : & (1 - \underbrace{\underbrace{0.9}_{\text{chance}_{DA}=1-0.1}}_{\underbrace{\underbrace{0.9}_{\text{chance}_{AB}=1-0.1}}_{\underbrace{\underbrace{0.9}_{\text{chance}_{BC}=1-0.1}}_{\underbrace{\underbrace{0.9}_{\text{chance}_{CD}=1-0.1}}_{0.6561}}}}) \times PV = 0.3439 \times PV \\
 DCBAD : & (1 - \underbrace{\underbrace{\underbrace{0.9}_{\text{chance}_{DC}=1-0.1}}_{\underbrace{\underbrace{0.9}_{\text{chance}_{CB}=1-0.1}}_{\underbrace{\underbrace{0.9}_{\text{chance}_{BA}=1-0.1}}_{\underbrace{\underbrace{0.9}_{\text{chance}_{AD}=1-0.1}}_{0.6561}}}})}_{0.6561}) \times PV = 0.3439 \times PV
 \end{aligned}$$

Furthermore, for a given cost of PD per unit of commodity, the expected cargo loss on route is:

$$\begin{aligned}
 DABCD : & \underbrace{\underbrace{\underbrace{\underbrace{\text{chance}_{DA} = 1 - 0.1}_{[A((1-0.9) \times PD)] +}}_{\underbrace{\text{chance}_{AB} = 1 - 0.1}_{B((1-0.81) \times 2PD) +}}_{\underbrace{\text{chance}_{BC} = 1 - 0.1}_{C((1-0.729) \times 3PD)] =}}_{= (0.1 + 0.38 + 0.813) \times PD = 1.293 \times PD} \\
 & \underbrace{\underbrace{\underbrace{\underbrace{\text{chance}_{DA} = 1 - 0.1}_{[A((1-0.9) \times PD)] +}}_{\underbrace{\text{chance}_{AB} = 1 - 0.1}_{B((1-0.81) \times 2PD) +}}_{\underbrace{\text{chance}_{BC} = 1 - 0.1}_{C((1-0.729) \times 3PD)] =}}_{= (0.1 + 0.38 + 0.813) \times PD = 1.293 \times PD}
 \end{aligned}$$

$$\begin{aligned}
DCBAD : & \underbrace{(chance_{DC} = 1 - 0.1) \times (chance_{CB} = 1 - 0.1) \times (chance_{BA} = 1 - 0.1)}_{[C((1-0.9) \times 3PD) +} \\
& \underbrace{B((1-0.81) \times 2PD) +}_{A((1-0.729) \times PD)] =} \\
& \underbrace{= (0.3 + 0.38 + 0.271) \times PD = 0.951 \times PD}
\end{aligned}$$

The total expected monetary cost (EMC) including the routing cost and the potential cost of expected vehicle and cargo loss is:

$$EMC_{DABCD} = 8 + 0.3439 \times PV + 1.293 \times PD \quad (6.1)$$

$$EMC_{DCBAD} = 8 + 0.3439 \times PV + 0.951 \times PD \quad (6.2)$$

Since the routing cost and the potential cost of vehicle loss is the same for both routes, route DCBAD is less costly ($EMC_{DCBAD} < EMC_{DABCD}$) because it has a lower potential cargo loss.

6.2 Problem Modelling

The elaborated model is building on top of previous initiatives [37, 121] as follows. It extends the regular linear programming model for MDVRP [37] with supplementary decision variables and constraints inspired from [121] and specifically modified in order to augment the model with risk on route assessment features. The model captures the problem using related decision variables as well as parameters, which will be discussed momentarily. Figure 6.2 presents a 16 node running example based on the multi-depot VRP case study problem data presented in [126], where 3 depots are pre-established. In addition, we consider the availability of 3 vehicles for each depot in order to accommodate replanning (adaptive planning) requirements. The running example serves to convey the modelling strategy as well as to subsequently conduct an illustrative case study in low and high risk settings. The figure depicts for each node its demand (or lack thereof in the case of depots) along with the corresponding 2D (abscissa, ordinate) coordinates.

The problem (depicted in Figure 6.2 (a) in the running example) is considered with respect to a set of vertices $V = \{1 \dots n\}$ (in the example $n = 16$ and $V = \{1, 2, \dots, 16\}$), which includes the depot nodes ($\{1, 2, 3\}$ in the example) and the customer nodes ($\{4, 5, \dots, 16\}$ in the example). For each node, there are specific locations (e.g., 1: 300,400 in the example) and based on the 2D

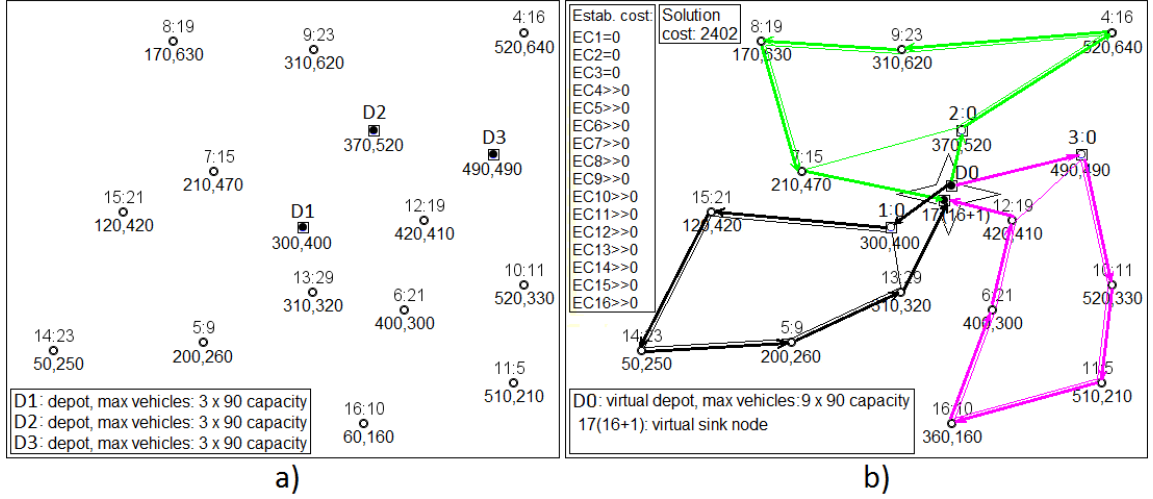


Figure 6.2: Running example with 3 depots and 13 demand nodes (a) and solution illustration with virtual nodes D0 and 17 as depot node and sink node for vehicle routes (b)

coordinates of the node locations, the routing cost c_{ij} can be determined across each edge $\langle i, j \rangle$ of the underlying full mesh graph $G = (V, E)$.

The problem has two perspectives as follows. We have the initial planning of risk constrained vehicle routes from multiple depots in order to serve all customer demands d_i (in the running example, $d_4 = 16$, $d_5 = 9$, ... $d_{16} = 10$ while $d_1 = d_2 = d_3 = 0$). Each depot has a maximum number of vehicles that can be used (3 in the example) of predefined cargo capacity (90 cargo units each in the example). The routes traverse over a transport network captured by the set of edges $E = (V \times V)$, where each edge has an associated risk probability ρ_{ij} of vehicle failure upon traversal. In addition, a maximum risk on route threshold can be specified for each vehicle tour. We also have the replanning perspective during the execution of the plan, in response changes in the risk probabilities on the transport network, risk on route threshold and/or vehicle failure events. This requires the generation of an updated problem to be solved where the vehicles that left their depots may have reduced amount of cargo (depending on deliveries made) while being considered to start from an on-the-route location. Likewise, the total customer demands may be lower if some deliveries have already been made before the initiation of replanning.

In order to capture the presence of multiple depots in the model, the strategy involves, as depicted in Figure 6.2 (b), the use of a virtual depot, namely node 0 (D0 in the running example)

in conjunction with the specification of establishment cost values (ec_i) for each non-virtual node i where a depot can be potentially located. In this setting, the first visited node i by any vehicle k starting from virtual node 0, represents the actual depot of k (captured by related decision variable w_i), with no routing cost for the first movement leg of vehicle k . In addition, a virtual sink node ($n + 1 = 17$ in the running example) is used to capture the return of the vehicles to their depots whereby the last leg movement routing cost (captured using related decision variables z_{ijk}) for every vehicle k is that of returning to the first visited node.

In order to more effectively convey the concept, Figure 6.2 (b) illustrates the solution (depicted with thin lines) with cost 2402, obtained when no risk is present. The figure depicts the virtual nodes (0 and $n+1$) enclosed in a four pointed star to indicate that there is no need to specify location coordinates for them. The solution actually results from 3 vehicle routes (depicted with thick lines), one from each depot, each starting at node 0 and ending at the sink node ($n + 1$):

First route:	$D0 \rightarrow \underbrace{1}_{\text{depot}} \rightarrow 15 \rightarrow 14 \rightarrow 5 \rightarrow 13 \rightarrow \underbrace{17}_{\text{depot D1}}$ ($D0 \rightarrow 1$ routing cost is 0 while routing cost $13 \rightarrow 17$ is given by $c_{13,1}$);
Second route:	$D0 \rightarrow \underbrace{2}_{\text{depot}} \rightarrow 4 \rightarrow 9 \rightarrow 8 \rightarrow 7 \rightarrow \underbrace{17}_{\text{depot D2}}$ ($D0 \rightarrow 2$ routing cost is 0 while routing cost $7 \rightarrow 17$ is given by $c_{7,2}$);
Third route:	$D0 \rightarrow \underbrace{3}_{\text{depot}} \rightarrow 10 \rightarrow 11 \rightarrow 16 \rightarrow 6 \rightarrow 12 \rightarrow \underbrace{17}_{\text{depot D3}}$ ($D0 \rightarrow 3$ routing cost is 0 while routing cost $12 \rightarrow 17$ is given by $c_{12,3}$).

The model has specific constraints, subsequently discussed, whereby the establishment cost of each established depot is considered only one time irrespective of how many vehicles have a particular depot node as the first visited node.

This strategy allows to capture the potential cost of establishing depots at the planning stage. However, when the problem involves already established depots as in the case of replanning, the establishment cost can be set to 0 for the depots while setting all the other establishment costs to a sufficiently large value. Moreover, in such situation, the decision variables capturing depot establishment can be turned into parameters.

6.2.1 Variables and Parameters

The following parameters are used in the model:

- $maxD$ stands for the maximum number of depots.
- $EC = \{ec_1 \cdots ec_n\}$ is the set of the establishment costs per potential depot.
- $MaxV = \{maxV_1 \cdots maxV_n\}$ is the set with the maximum vehicles per potential depot.
- $vLimit$ represents the maximum number of vehicles that can be used.
- $K = \{1 \cdots vLimit\}$ represents the set of vehicles.
- Q_k ($k \in K$) represents the cargo capacity of the vehicles.
- M represents a sufficiently large number.
- PV represents per unit vehicle cost.
- PD represents per unit cargo cost.
- $maxRisk$ represents the maximum vehicle risk on route.
- c_{ij} represents the routing cost of traversing edge $\langle i, j \rangle$.
- ρ_{ij} represents the risk of traversing edge $\langle i, j \rangle$.
- d_i represents the demand at node i .

The model employs the following decision variables:

- $x_{ijk} \in \{0, 1\}$: boolean variables to determine the routes (1, if edge $\langle i, j \rangle$ is taken by vehicle k ; 0 otherwise) where $i, j \in V \cup \{0, n + 1\}$ and $k \in K$.
- $z_{ijk} \in \{0, 1\}$: boolean variables to determine route completion (1, if i and j are respectively the last served node and the depot node of vehicle k ; 0 otherwise) where $i, j \in V$ and $k \in K$.
- $w_i \in \{0, 1\}$: boolean variables for depot selection (1 if i is a depot; 0 otherwise) where $i \in V$.
- $y_i \in [0, 1]$: real number for the risk of reaching node i by any vehicle, bounded by $maxRisk$, where $i \in V$.
- $v_k \in [0, 1]$: real number for the tour risk of vehicle k , bounded by $maxRisk$, where $k \in K$.

6.2.2 Non-Linearity Aspect

As part of the modelling process, decision variables (y_i and v_k) are employed to identify the risk of serving each demand node and to determine the risk of vehicle loss on the route respectively. As discussed in Section 6.1.3, the chance of a vehicle to reach a particular node, is reducing in a progressive manner, with each successively traversed routing leg exhibiting non-zero risk. Conversely, the risk gets progressively larger along the vehicle route. For instance, taking from the example in Section 6.1.3 a sub-route $D \rightarrow A \rightarrow B \rightarrow C$, the risk of the vehicle to arrive at node C is $y_C = 1 - (1 - y_B) \times (1 - \rho_{BC})$ where $y_B = 1 - (1 - y_A) \times (1 - \rho_{AB})$ and $y_A = \rho_{DA}$.

Thus, a route taken over a particular sequence of nodes directly impacts the determination of risk associated to each node on that route. In the proposed RCMDVRP setting, variables x_{ijk} and z_{ijk} determine optimal vehicle routes. Alongside, y_i and v_k are related to x_{ijk} and z_{ijk} as follows:

$$y_j \geq (1 - (1 - y_i) \times (1 - \rho_{ij})) \times \sum_{k \in K} x_{ijk}, \forall i \in V, j \in V, i \neq j \quad (6.3)$$

$$v_k \geq \sum_{j \in V} (1 - (1 - y_i) \times (1 - \rho_{ij})) \times z_{ijk}, \forall k \in K, \forall i \in V, i \neq j \quad (6.4)$$

Eqn. (6.3) indicates that the risk of serving demand node j , namely y_j is always positive and it increases from the risk of the previous visited node i along the path of a visiting vehicle k , depending on the risk value ρ_{ij} of the leg segment $\langle i, j \rangle$, as determined by x_{ijk} . Likewise, Eqn. (6.4) indicates that the overall tour risk of vehicle k , namely v_k is also positive and determined from the risk of the last visited demand node (y_i) by vehicle k and the risk value ρ_{ij} to reach depot j , as determined by variable z_{ijk} . One can observe that both Eqn. (6.3) and Eqn. (6.4) impose non-linear constraints over y_i and v_k , which may pose convexity issues for the corresponding model and limit the usability of linear solvers such as CPLEX¹ (used to exercise the model). The non-linearity can be avoided by using additive risk approximation for small risk probability values over the tour leg segments, as detailed in [49]. Accordingly, Figure 6.3, depicts the effect of additive approximation for increasingly larger tour leg risk values, ranging from 1% to 5%. As such, the figure shows 5 pairs of data series. These series illustrate an increasing overestimation gap between the additive risk approximation and the actual cumulative risk probability for increasingly larger sequential leg count values, up to 10% gap between the approximated and actual cumulative risk probability.

¹LOG CPLEX 12.7.0 user's manual, 2016, IBM.

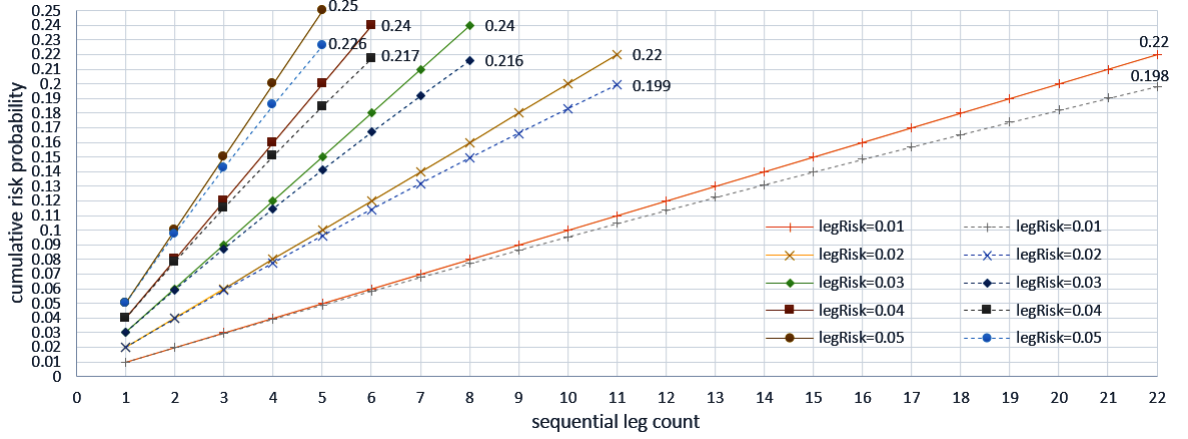


Figure 6.3: Additive risk approximation (solid line) vs. actual cumulative risk (dashed line)

It is worthy to mention that according to the statistic results provided by AAA Foundation for Traffic Safety², the risk probability of on-road vehicle accident in a non-hostile environment is as small as 0.00002 per km, which corresponds to around 0.002 per 100 km. As such, one can effectively employ additive risk approximation even in hostile environments where one could expect a 10 fold higher (relative to a non-hostile environment situation) risk probability of 0.02 for a vehicle movement leg of 100 km.

6.2.3 Linear Programming Model using Additive Risk Approximation

We present next the linear programming model formulation as follows:

$$\begin{aligned}
 \min \sum_{i \in V} ec_i \times w_i + \sum_{i \in V} \sum_{j \in V, i \neq j} c_{ij} \sum_{k \in K} x_{ijk} + \sum_{i \in V} \sum_{j \in V, i \neq j} c_{ij} \sum_{k \in K} z_{ijk} \\
 + PD \times \sum_{j \in V} d_j y_j + PV \times \sum_{k \in K} v_k
 \end{aligned} \tag{6.5}$$

Subject to:

Flow Conservation:

$$\sum_{k \in K} \sum_{i \in V} x_{ijk} = 1 - w_j, \forall j \in V, i \neq j \tag{6.6}$$

²<https://aaafoundation.org>, Last accessed on 1 March 2019.

$$\sum_{k \in K} \sum_{i \in V} x_{iik} = 0 \quad (6.7)$$

$$\sum_{j \in V} x_{0jk} = \sum_{j \in V} x_{j(n+1)k}, \forall k \in K \quad (6.8)$$

$$\sum_{k \in K} x_{0(n+1)k} = 0 \quad (6.9)$$

$$\sum_{k \in K} \sum_{j \in V \cup \{0\}} x_{(n+1)jk} = 0 \quad (6.10)$$

$$\sum_{k \in K} \sum_{i \in V \cup \{n+1\}} x_{i0k} = 0 \quad (6.11)$$

$$\sum_{i \in V \cup \{0\}} x_{ihk} - \sum_{j \in V \cup \{n+1\}} x_{hjk} = 0, \forall k \in K, h \in V, i \neq h, j \neq h \quad (6.12)$$

The flow conservation constraints impose specific restrictions on the vehicle routes as follows. Eqn. (6.6) states that customer nodes are visited only once while precluding the vehicles from visiting the established depots from the customer nodes. Eqn. (6.7) enforces that a vehicle route is a sequence of edges where each edge starts and ends at different nodes. Eqn. (6.8) assures that each vehicle that starts from the virtual depot node 0 must also reach the sink node $n + 1$. Furthermore, Eqn. (6.9) imposes that a vehicle cannot directly move from the virtual depot node to the sink node while Eqn. (6.10) and Eqn. (6.11) restrict respectively that no vehicle can move from the sink node and that no vehicle can revisit the the virtual depot. Eqn. (6.12) assures that any vehicle reaching a demand node must also leave it. Together, Eqns. (6.9)-(6.12) restrict that a vehicle may start a tour only from the virtual depot 0.

Depot Establishment:

$$\sum_{k \in K} x_{0jk} \geq w_j, \forall j \in V \quad (6.13)$$

$$x_{0jk} \leq w_j, \forall j \in V, \forall k \in K \quad (6.14)$$

$$\sum_{i \in V} w_i \leq \max D \quad (6.15)$$

Regarding the depot establishment constraints, Eqn. (6.13) ensures that there is at least one vehicle per established depot. Eqn. (6.14) affirms that all vehicles first visit an established depot location. Alongside, Eqn. (6.15) restricts the maximum number of depots that can be established.

Maximum Vehicle Restriction:

$$\sum_{k \in K} x_{0jk} \leq \max V_j \times w_j, \forall j \in V \quad (6.16)$$

$$\sum_{k \in K} \sum_{j \in V} x_{0jk} \leq |K| \quad (6.17)$$

With respect to the maximum vehicle restriction constraints, as per Eqn. (6.16), the total number of vehicles per depot cannot not exceed limit $\max V_j$ if a depot is established at node j . Also, Eqn. (6.17) limits to $|K|$ the total number of available vehicles.

Capacity Restriction:

$$\sum_{j \in V} d_j \sum_{i \in V, i \neq j} x_{ijk} \leq Q_k, \forall k \in K \quad (6.18)$$

The capacity restriction constraint, as denoted by Eqn. (6.18), restricts the total serving in a tour for each vehicle k , to a maximum of Q_k . However, the demands of the nodes where depots are established are to be served by the respective depots without any vehicle use, as imposed by Eqn. (6.6) and Eqn. (6.18).

Sub-tour Elimination:

$$\sum_{i \in S} \sum_{j \in S} x_{ijk} \leq |S| - 1; \forall S \subseteq V \cup \{0, n+1\}, |S| \geq 2, \forall k \in K \quad (6.19)$$

Eqn. (6.19) represents a modified version of the generalized sub-tour elimination constraint from [142], ensuring that sub-tours cannot form in the solution.

Auxiliary End of Tour:

$$\sum_{i \in V} \sum_{j \in V} z_{ijk} = \sum_{j \in V} x_{0jk}, \forall k \in K \quad (6.20)$$

$$x_{0jk} + x_{i(n+1)k} \leq 1 + z_{ijk}, \forall k \in K, i \in V, j \in V \quad (6.21)$$

These constraints set variables z_{ijk} for the end of tour of each serving vehicle k . Eqn. (6.20) and Eqn. (6.21) track the edge between vehicle's last visited node and the assigned depot of the vehicle in order to derive the appropriate return cost.

Risk Evaluation:

$$y_0 = 0 \quad (6.22)$$

$$y_i \leq M \times (1 - w_i), \forall i \in V \quad (6.23)$$

$$y_j \geq y_i - M + (\rho_{ij} + M) \times \sum_{k \in K} x_{ijk}, \forall i \in V, j \in V, i \neq j \quad (6.24)$$

$$y_i + \rho_{ij} \times \sum_{k \in K} z_{ijk} \leq \max Risk, \forall i \in V, j \in V, i \neq j \quad (6.25)$$

$$v_k \geq y_i - M + \sum_{j \in V} (\rho_{ij} + M) \times z_{ijk}, \forall k \in K, \forall i \in V \quad (6.26)$$

Eqns. (6.22)-(6.26) capture via y_i the additive risk for visiting each demand node i . Thus, Eqn. (6.22) and Eqn. (6.23) specify that risk is not considered at virtual node (0) nor at all established depots respectively. In contrast, the value of risk progressively increases (depending on ρ_{ij}) along the route as per Eqn. (6.24). M represents a sufficiently large constant used to assure that risk values are unrelated for reaching two different demand nodes, except when a vehicle tour connects them. Eqn. (6.25) imposes the maximum bound of risk for every node and vehicle tour in relation to the application of the additive risk increase. Finally, Eqn. (6.26) captures, via variables v_k , the overall risk of each vehicle tour, thereby ascertaining the corresponding vehicle loss risk.

Thus, the objective function in Eqn. (6.5) minimizes the overall depot establishment cost along with the overall routing cost, overall expected vehicle loss cost and overall expected cargo loss cost. The model is subjected to seven categories of constraints: *Flow Conservation*, *Depot Establishment*, *Maximum Vehicle Restriction*, *Capacity Restriction*, *Sub-tour Elimination*, *Auxiliary End of Tour* and *Risk Evaluation*.

6.3 Solution Approach

The proposed solution approach extends the multi-point insertion cost gradient descent heuristic technique introduced in [132]. The latter can handle different problem variants from the vehicle routing family (in traditional setting, without risk on route), being aimed at MDVRP with split delivery. In this regard, an in-depth characterization of its features and performance is presented in [120]. A noteworthy feature of the solution search mechanism is that it can be decentralized across several computing nodes that can belong to different decision making participants (as it employs a seed-based pseudo-random number generator used to steer the solution search). This allows for collaborative solving of logistic support routing problems by a team of decision making participants typically located at the depots. Distributed collaborative solution generation without sharing fleet information is also possible, as shown in [131].

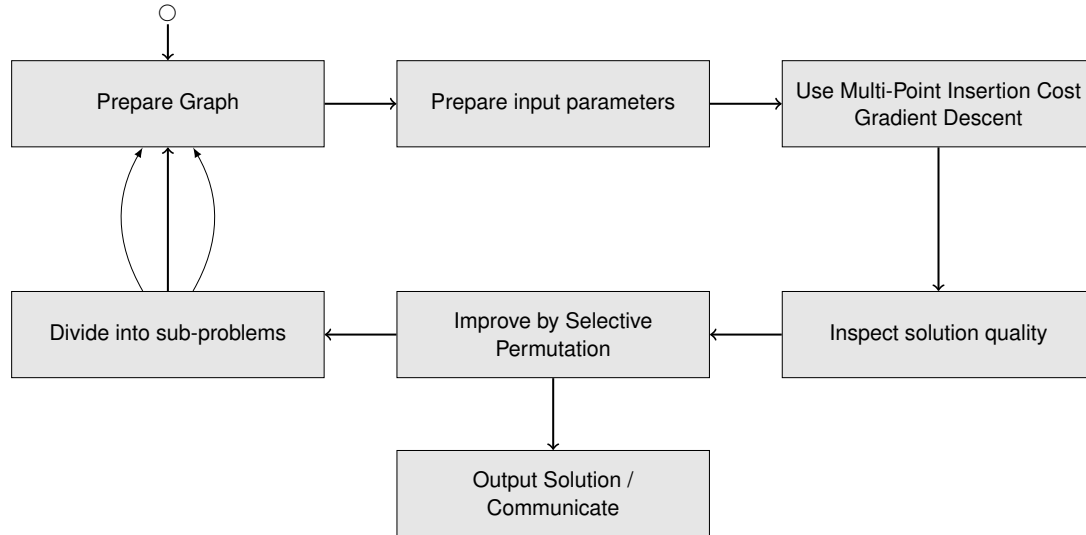


Figure 6.4: Solution generation procedure

In the context of the RCMDVRP, the technique is extended with risk mitigation features but without split delivery considerations in order to limit the vehicle risk exposure. Figure 6.4 presents the synopsis of the solution generation approach, which assures that a solution is ready for each participant after the first pass. The participants can solve a common problem instance using the heuristic technique and possible refinements during a dynamic solution exploration process. In the latter, progressively better bounds can be exchanged, leading to a near optimal solution communicated among participants. In a likewise manner, if needed, plan alternatives can be calculated in a distributed manner before and during the execution of the plan. The alternatives present possible course of action for various circumstances. This can include the consideration of contingency options for potential vehicle losses or alternative routes for risk factors that can potentially change. The extended multi-point insertion cost gradient descent algorithm is presented next.

6.3.1 Algorithm

The algorithm for solving RCMDVRP includes a graph preparation procedure involving the substitution of the transport network graph G , with a combined cost traversal sorted map G_{map} based on the routing cost values given by c_{ij} and the risk traversal values given by ρ_{ij} . The map provides for each node $i \in V$, an ordered sequence with the combined traversal and risk-related cost of directly reaching i from the other nodes in $V \setminus \{i\}$. The set of input parameters is as follows:

- *Initial solution bound (S_{nn})*: Provides the initial upper bound used during the search. Usually, it is the nearest neighbour solution that can be generated very fast.
- *Search map (s_{map})*: Ascending sorted map of solution fragments based on the routing and risk-related cost of each fragment such that the lowest cost solution fragment is placed at the head (top). The other solution fragments are placed in subsequent positions in ascendant manner (based on their respective cost). Its maximum size is limited with parameter $sMapLmt$ and it is initialized with the unexpanded multi-tour solution fragment having the available vehicles, with full capacity, located at their depots.
- *Serve set ($serve_{set}$)*: Array where each entry contains an ordered set (of parameterized maximum cardinality $serveSetLmt$) containing solution fragment cost values corresponding to their fraction of the total serving in a complete solution. Once the maximum cardinality is

reached, any subsequent candidate fragment cost value can only enter in the corresponding set if its cost is lower than the existing maximum value of the set and if so by eliminating the existing maximum value first. The array can be empty upon initialization or left populated from a previous search performed with a different seed.

- *Maximum neighbours (maxnbr)*: Parameter indicating that during solution fragment exploration, any selected vehicle in a solution fragment can be used to serve customers only among its nearest (in terms of cost) *maxnbr* neighbours.
- *Randomizing seed (seed)*: Unique number used to generate repeatable (for the same seed value) pseudo-random choices, allowing for stochastic candidate vehicle selection during solution fragment exploration. Initialized with an user specified integer value.
- *Maximum steps (maxStep)*: user specified integer value for auxiliary stop condition.

Algorithm 4 depicts the risk mitigation extension of the solution algorithm presented in [132]. The algorithm is presented from a higher level of abstraction in order to provide a more effective conceptual understanding, while making use of pseudocode and self explanatory operations such as Get, Next, etc. (the operations names have capitalized first letter). At line 1, the solving procedure begins using the the specified inputs parameters. At line 2, we have the global knowledge needed for solution generation. At line 3, the procedure initializes the step counter *stepCnt*, total demand D^* and current best solution S^* . A while loop starts at line 4 for as long as s_{map} is not empty and *stepCnt* > 0 . At line 5, *stepCnt* is decremented. A multi-tour s is popped from the head (top) of s_{map} at line 6. At lines 7 to 9, s is checked if it includes more than one tour and if so, the list of tours in s is shuffled. A for loop starts at line 10 over the list of tours in s . At line 11, *maxNN* is initialized. A while loop starts at line 12 for as long as *maxNN* > 0 and the size of s_{map} is under *sMapLmt*. At line 13, the next potential destination node is derived (relative to the last served node of the selected tour) along with its serving need (line 14). The possibility of serving the node by the selected tour is checked at line 15, and if so, the node is inserted in the selected tour at line 16. Lines 17 to 19 check if the cost of s exceeds that of S^* in order to continue the loop from its beginning. Lines 20 to 24 check if all demands have been served, thereby completing a solution, and if so, a node swap cost improvement is attempted followed by the corresponding

Algorithm 4 Heuristic Search (HS) for RCMDVRP near-optimal vehicle routes

```
1: Inputs:  $S_{nn}$ ,  $s_{map}$ ,  $sMapLmt$ ,  $serve_{set}$ ,  $serveSetLmt$ ,  $maxnbr$ ,  $seed$ ,  $maxStep$ ,  $maxVehRisk$ 
2: Global knowledge:  $G_{map}$ ,  $d_i$ ,  $c_{ij}$ ,  $\rho_{ij}$ 
3: Initially:  $stepCnt \leftarrow maxStep$ ;  $D^* \leftarrow TotalDemand(d_i)$ ;  $S^* \leftarrow S_{nn}$ ;
4: while  $s_{map}$  is not empty and  $stepCnt > 0$  do
5:    $stepCnt \leftarrow stepCnt - 1$ ;
6:   Pop MultiTour  $s$  from top of  $s_{map}$ 
7:   if  $s$  contains more than one tour then
8:     Shuffle  $s$  using  $seed$ ;
9:   end if
10:  for each  $selectedTour$  in  $s$  do
11:     $maxNN \leftarrow maxnbr$ ;
12:    while  $maxNN > 0$  and  $Size(s_{map}) < sMapLmt$  do
13:       $nextDst \leftarrow GetNextNeighbour(LastServedNode(selectedTour))$ ;
14:       $nextServeNeed \leftarrow GetDemand(nextDst)$ ;
15:      if  $nextServeNeed > 0$  and  $nextServeNeed \leq RemainingCapacity(selectedTour)$  then
16:         $InsertNode(selectedTour, nextDst, maxVehRisk, GetRisk(LastServedNode(selectedTour), nextDst))$ 
17:        if  $Cost(s) > Cost(S^*)$  then
18:          continue;
19:        end if
20:        if  $ServeAmount(s) = D^*$  then
21:           $swapTourSol \leftarrow LocalizedSwap(s)$ 
22:          if  $Cost(swapTourSol) < Cost(s)$  then
23:             $s \leftarrow swapTourSol$ ;
24:          end if
25:           $S^* \leftarrow s$ ;
26:          Remove all entries from  $s_{map}$  with cost greater than  $Cost(S^*)$ ; continue;
27:        end if
28:        if  $Size(serve_{set}(serveAmount(s))) = serveSetLmt$  and  $Cost(s) \geq Max(serve_{set}(serveAmount(s)))$  then
29:          continue;
30:        else
31:          if  $Size(serve_{set}(ServeAmount(s))) = serveSetLmt$  then
32:            Remove  $Max(serve_{set}(serveAmount(s)))$  from  $serve_{set}(serveAmount(s))$ 
33:          end if
34:          Put  $Cost(s)$  in  $serve_{set}(serveAmount(s))$ ;
35:        end if
36:        Put  $s$  in  $s_{map}$ 
37:      end if
38:       $maxNN \leftarrow maxNN - 1$ ;
39:    end while
40:  end for
41: end while
42: return  $S^*$ ;
```

update of s if it is the case. At lines 25 to 27, S^* is updated and all entries in s_{map} with higher cost than S^* are removed. Lines 28 to 35 check if the size of the set in $serve_{set}$ corresponding to the serving amount of s , reached $serveSetLmt$ and if the cost of s exceeds the maximum value in the set, in order to continue the loop from its beginning if it is the case; otherwise the maximum value in the set is removed and the corresponding set is updated with the cost corresponding to s . At lines 36 to 37, s is reinserted in s_{map} . At line 38, $maxNN$ is decremented. Lines 39 to 41 delimit the scope for their corresponding loops. The current best solution is returned at line 42.

Since the algorithm extends the one in [132], it is mainly based on the concept of multi-point stochastic insertion cost gradient descent. The multi-point aspect of the technique refers to the fact that solution fragments are constructed by inserting unserved nodes in multiple points corresponding to multiple selected vehicle tours. The stochastic aspect relates to using a seed-based pseudo-randomized vehicle selection for visiting the demand nodes. The insertion cost gradient descent can be seen in that the lower cost (combined routing and risk-related cost) fragments maintained in s_{map} are explored before the higher cost ones. This is due to the ascending cost ordering of the fragments in s_{map} whereby they are fetched from the head thereof. In this manner, the comparatively better (for a given amount of demand serving) fragments accumulate toward the head whereas the cost-wise uninteresting fragments (for a given amount of serving) accumulate toward the tail. The cost-wise uninteresting fragments can also be discarded once the size limit has been reached and their cost exceeds the cost of the last element in s_{map} .

6.3.2 Discussion

In the following, the procedure is further detailed in order to offer additional insights. The generation of connecting fragments has an important combinatorial aspect that may lead to a large number of possible fragments that need to be further explored. However, many such fragments are in essence a rearrangement of other potential fragments without a cost benefit. Consequently, the algorithm uses parameter $sMapLmt$ to limit the size of s_{map} such that it discards the cost-wise uninteresting expanded fragments, thereby adjusting the depth search of the procedure. A larger value of $sMapLmt$ corresponds to expanding a larger number of fragments, which leads in general to better solutions at the expense of using more computational resources (both time and memory space). Conversely, a smaller value of $sMapLmt$ would lead in general to less cost effective

solutions that are obtained more quickly and with a smaller memory footprint. With respect to the initial near-neighbour solution S_{nn} , it is derived by employing the algorithm with $maxNbr = 1$ and setting the initial cost of S_{nn} to an arbitrary high value. This allows to have an initial upper bound feasible solution. The value of the latter represents the initial value of the “best solution so far” reference, relative to which the subsequently identified solutions should have a lower cost.

After obtaining the initial feasible solution, the algorithm can be fully employed. In essence, it repeatedly explores (while loop between lines 4 to 41) multi-point solution fragments (fractional serving of the total demand level) with a bias toward the least costly connections in G_{map} . In this context, after a multi-tour is popped from the top of s_{map} (line 6), the seed-based pseudo-random number generator shuffles (line 8) the selection of the vehicles that may be able to visit the node considered to be served next. The latter is considered among a maximum of $maxnbr$ nodes, provided that the size of s_{map} is less than $sMapLmt$ (lines 12 to 14). The `GetNextNeighbour` routine provides the next potential destination ($nextDst$) from G_{map} with respect to the node provided by the routine `LastServedNode` over the selected tour. The `GetDemand` routine provides from d_i the corresponding demand to be served ($nextServeNeed$).

The fragments are maintained in s_{map} based on their respective cost. The popped fragments from s_{map} are explored from the top for further expansion and potential reinsertion in s_{map} . Each popped solution fragment (line 6) can be expanded, in various ways, by inserting additional unserved nodes in a selected tour (line 16), until the expanded fragment s either becomes a complete solution or it is discarded as being too costly.

The `InsertNode` routine updates the cumulative vehicle risk of the *selectedTour* for serving $nextDst$. It uses the previous cumulative vehicle risk (corresponding to the previously inserted node or 0 when no node was inserted before) and the risk value provided by the `GetRisk` routine. The latter provides from ρ_{ij} the corresponding traversal risk between the node provided by the `LastServedNode` routine and $nextDst$ node. In addition, the `InsertNode` routine updates as well the vehicle tour risk (the cumulative risk of returning to its depot). All updates use the actual cumulative risk estimation since the heuristic technique has no limitation that would require the use of the additive risk approximation. The `InsertNode` routine also updates the combined routing cost (based on the corresponding values from c_{ij}) and risk-related cost for the *selectedTour*, which in turn leads to a cost update for s . In the case where the updated cumulative tour risk for the

selectedTour exceeds the maximum vehicle risk *maxVehRisk*, the combined cost incurs a multiplicative penalty in order to make the fragment less competitive and more likely to be discarded.

In essence, the expansion (line 16) and reinsertion (line 36) proceeds as long as the fragment cost is less than the cost of the best solution so far, otherwise (line 18) the not qualifying fragment is discarded (the application of supplementary qualifying conditions based on the amount of serving will be discussed momentarily). Also, whenever the expansion of a fragment results in all demands being served, no reinsertion is performed since a complete solution is formed. In addition, no reinsertion is performed when a fragment expansion results in a situation where none of the vehicles have enough remaining usable capacity to serve any of the unserved nodes. Each formed solution can potentially be improved by attempting a localized node swapping (permutation across adjacent nodes) over each tour of multi-tour s (line 21). In the case where some tours exceed the maximum vehicle risk before node swapping, this may result in bringing the maximum vehicle risk to a value less or equal to *maxVehRisk*, thereby obtaining for such tours an important cost benefit by avoiding the penalty caused by exceeding *maxVehRisk*. Subsequently, the solution is compared against the best solution so far. The latter is also updated whenever a better solution is identified (lines 22 to 24). In addition, all cost fragments in the queue with higher cost than the best solution so far are discarded (line 26) since a complete solution with lower cost already exists.

Furthermore, to apply selective pressure on fragment expansion, the algorithm uses *serve_{set}* to maintain ordered serving sets with parameterized maximum cardinality corresponding to fractional levels of demand served. Whenever the maximum cardinality is reached for such set, the only allowed fragments to expand are those that qualify under the maximum cost value of that set (for that particular fractional serving). If it is the case, the maximum cost value is removed from the set before inserting the value corresponding to the qualifying fragment (lines 28 to 35). This lowers the maximum cost value in the set to the next higher value remaining therein, thus applying further pressure on fragment selection. The *serve_{set}* is essentially used to build up the knowledge related to the specific topology and serving availability characterizing the problem instance that is solved. The knowledge gathered is captured by the adjustable cost bounds corresponding to particular fractions of total demand, as discovered during fragment exploration. This knowledge allows the procedure to be more effectively guided by qualifying or disqualifying potential fragments while they are being expanded.

Finally, the looping procedure of expansion and reinsertion of solution fragments ends by means of two stop conditions (primary and auxiliary). The primary stop condition is met when no solution fragment remains to be explored (s_{map} is empty) whereas the auxiliary stop condition is met when the step counter $stepCnt$ reaches 0. The latter stop condition allows the algorithm to finish within a predictable time interval (maximum number of loops) for larger values of $sMapLmt$. Thereafter, the best solution so far is returned as the best solution obtained (line 42).

The approach also allows for sub-problem solution refinement as follows. An obtained solution can potentially be improved by non-deterministically delineating sub-problems in the form of tour pairs, in pursuit of cost improvement. This allows to divide the initial problem into smaller sub-problems for iterative cost savings by successively re-applying the algorithm on the sub-problems for which cost savings can be achieved, as illustrated in Figure 6.4. In this context, when a cost saving is obtained for a sub-problem, the other sub-problems sharing a tour with the sub-problem just improved, are reconsidered for improvement. In this manner, better solutions can be progressively found while reducing the number of sub-problems reconsidered for further improvement over multiple iterations, until no further cost savings can be obtained.

6.4 Case Study

We present next an illustrative two-part case study that highlights the effect of considering initially low on route risk values and then high on route risk values over the obtained RCMDVRP planning solutions. The case study is carried out over the running example problem depicted in Figure 6.2. As shown, the problem represents a small scale RCMDVRP with 16 nodes, 3 of which are pre-established depots. The demand nodes are shown by small circles while the depots are marked by squares. Below each node we have its location coordinates (abscissa, ordinate) while above we have the node id and its demand (id:demand), or the depot label and the id (D1, D2, D3). Moreover, each depot has a maximum of 3 vehicles, each with a capacity of 90 cargo units.

In order to illustrate the effect of risk on route, we solve the problem for a given risk of travel while considering various combinations of unit vehicle cost and unit cargo cost. While the approach allows for the possibility of considering specific risk traversal values between any nodes of the problem, without loss of generality, the risk traversal values considered in the case study

are uniform, for a more effective presentation of the results. In this setting, the use of uniform risk values for every edge of the problem (i.e., every potential vehicle tour leg), allows to better convey the intricacies of the approach and to more meaningfully compare the various obtained solutions.

Thus, we initially consider low risk values for edge traversal (0.01 and 0.02) with a maximum vehicle risk of 0.1, followed by the consideration of high risk values for edge traversal (0.1 and 0.2) with a maximum vehicle risk of 0.5. In each case we use the values of (0, 1000 and 10000) for the unit vehicle cost and (0, 10 and 100) for the unit cargo cost. We chose larger values for vehicle cost since a vehicle unit is typically more costly than a cargo unit. The 0 values are used to consider the effect of risk relative to: maximum vehicle risk and cargo only (unit vehicle cost is 0, unit cargo cost > 0), maximum vehicle risk and vehicle only (unit vehicle cost > 0 , unit cargo cost is 0) and maximum vehicle risk only (unit vehicle cost is 0, unit cargo cost is 0).

6.4.1 Route Generation on Low Risk of Vehicle Failure

In the first part of the case study that deals with low risk of vehicle failure, we examine the effect of two risk traversal values. The initial risk traversal value considered is 1% for each vehicle movement across every edge of the problem. Subsequently, a value of 2% is also considered. As shown in Figure 6.3, such low risk values allow the use of additive risk approximation whereby the cumulative vehicle risk, over successive routing segments (tour legs), can be approximated by the summation of the risk probability values of each tour leg. The approximation of cumulative risk is shown up to a deviation of 10% (approximated vs. actual cumulative risk). For the considered deviation, 1% leg risk holds for 22 tour legs while 2% risk holds for 11 tour legs.

The case study problem has 13 demand nodes (4 to 16), which means that for a sufficiently large vehicle capacity, a tour cannot exceed 14 legs. However, for the capacity of 90 cargo units, a vehicle tour cannot exceed 8 legs (serving at most 7 nodes) since the 7 nodes with the smallest demand are 11:5, 5:9, 16:10, 10:11, 7:15, 4:16, 8:19, which require $5+9+10+11+15+16+19=85$ cargo capacity. Since the additive risk approximation allows to model the problem in linear fashion, we can assess the effectiveness of the heuristic technique relative to the correspondingly obtained case study solutions.

Considering 3 values for the unit vehicle cost and 3 values for the unit cargo cost results in a total of 9 combinations. Figure 6.5 depicts the corresponding routes obtained for tour leg traversal

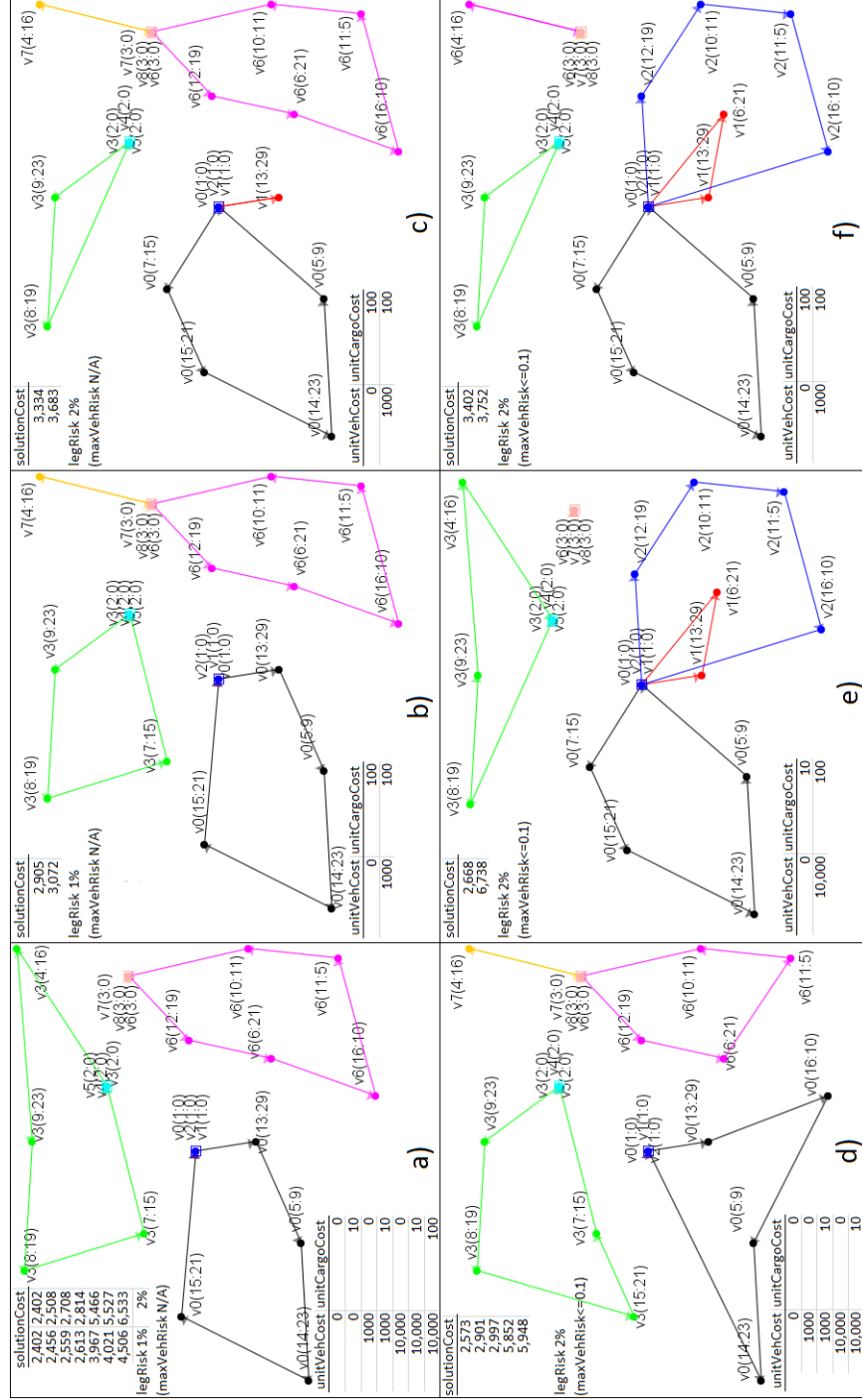


Figure 6.5: Risk on route solution sensitivity (1% leg risk, without max. vehicle risk, solutions still under threshold; 2% leg risk, without and with max. vehicle risk)

risk values of 1% and respectively 2% without and with the consideration of maximum vehicle risk. Since the leg risk values are small, there are several <unit vehicle cost, unit cargo cost> combinations for which the corresponding solutions share the same routes while having different overall cost. Figure 6.5(a) depicts the routes corresponding to <unit vehicle cost, unit cargo cost> combinations <0, 0>, <0, 10>, <1000, 0>, <1000, 10>, <10000, 0>, <10000, 10>, <10000, 100> both for 1% and 2% leg risk when the maximum vehicle risk is not applied. For the combination <0, 0> in particular, the solution takes into account just the routing cost. As such it represents the same solution obtained when no risk is considered and therefore also represents the solution of the original MDVRP problem. Figure 6.5(b) depicts the routes for the combinations <0, 100>, <1000, 100> for 1% leg risk when the maximum vehicle risk is not applied while Figure 6.5(c) depicts the routes corresponding to combinations <0, 100>, <1000, 100> for 2% leg risk when the maximum vehicle risk is not applied.

Figure 6.5(d) depicts the routes corresponding to the combinations <0, 0>, <1000, 0>, <1000, 10>, <10000, 0>, <10000, 10> for 2% leg risk when the maximum vehicle risk is 10%. Figure 6.5(e) depicts the routes corresponding to the combinations <0, 10>, <10000, 100> for 2% leg risk when the maximum vehicle risk is 10%. Figure 6.5(f) depicts the routes corresponding to the combinations <0, 100>, <1000, 100> for 2% leg risk when the maximum vehicle risk is 10%.

Figure 6.6 provides a spreadsheet with the details for all solutions depicted in Figure 6.5, including the details for the solutions obtained with a restriction of maximum vehicle risk on route of 10%. The spreadsheet has 8 columns corresponding in the listed order to the following metadata: leg risk/(maximum vehicle risk); unit vehicle cost; unit cargo cost; solution vehicle count; vehicle risk (minimum values to maximum values); average vehicle risk; solution cost; routing cost. The values obtained using the additive risk approximation are provided inside square brackets on the appropriate columns. The first line of the spreadsheet lists the aforementioned metadata. The spreadsheet then provides the solution details for the various <unit vehicle cost, unit cargo cost> combinations and leg risk values. We note that for 1% leg risk, the maximum vehicle risk can be observed for vehicle v6 (5.85% and respectively 6% on additive approximation) across all <unit vehicle cost, unit cargo cost> combinations, which is below the 10% threshold. Also, the solutions obtained with additive risk approximation have a little higher cost since the approximation slightly overestimates the cumulative risk. However, the corresponding routes and

legRisk	unitVehCost	unitCargoCost	solVehCnt	vehRisk (min to max) [additive risk approx.]		averageVehRisk [additive risk approx.]	solCost [additive risk approx.]	routingCost
legRisk=1% (maxVehRisk N/A)	0	0	3	v0, v3: v6:	4.90 [5.00] % 5.85 [6.00] %	5.22 [5.33]%	2,402 [2,402]	2,402
	0	10					2,456 [2,457]	
	1000	0					2,559 [2,562]	
	1000	10					2,613 [2,617]	
	10,000	0					3,967 [4,002]	
	10,000	10					4,021 [4,057]	
	10,000	100					4,506 [4,547]	
	0	100	4	v7:	1.99 [2.00]%	4.17 [4.25]%	2,905 [2,910]	2,422
	1000	100		v3:	3.94 [4.00]%		3,072 [3,080]	
				v0:	4.90 [5.00]%			
				v6:	5.85 [6.00]%			
legRisk=2% (maxVehRisk N/A)	0	0	3	v0, v3: v6:	9.61 [10.0] % 11.42 [12.0] %	10.21 [10.66] %	2,402 [2,402]	2,402
	0	10					2,508 [2,511]	
	1000	0					2,708 [2,722]	
	1000	10					2,814 [2,831]	
	10,000	0					5,466 [5,602]	
	10,000	10					5,527 [5,711]	
	10,000	100					6,533 [6,692]	
	0	100	5	v1, v7:	3.96 [4.00]%	6.97 [7.2]%	3,334 [3,349]	2,481
	1000	100		v3:	5.88 [6.00]%		3,683 [3,709]	
				v0:	9.61 [10.00]%			
				v6:	11.42 [12.00] %			
legRisk=2% (maxVehRisk<=0.1)	0	0	4	v7:	3.96 [4.00]%	8.20 [8.50]%	2,573 [2,573]	2,573
	1000	0		v0, v3, v6:	9.61 [10.00]%		2,901 [2,913]	
	10,000	0					5,852 [5,973]	
	1000	10		v7:	3.96 / - %	8.20 / 8.22%	2,997 / 2,997 5,948 / 5,954	2,573 / 2,581
	10,000	10		v1:	- / 5.88%			
				v3:	9.61 / 7.76%			
				v0:	9.61 / 9.61%			
				v2:	- / 9.61%			
				v6:	9.61 / - %			
	[1000]	[10]		v7:	[4.00] / - %	[8.50] / [8.50]%	[3,011] / [3,009] [6,071] / [6,069]	
	[10,000]	[10]		v1:	- / [6.00]%			
				v3:	[10.00] / [8.00]%			
				v0:	[10.00] / [10.00]%			
				v2:	- / [10.00]%			
				v6:	[10.00] / - %			
	0	10	5	v1:	5.88 [6.00]%	8.22 [8.50]%	2,668 [2,669]	2,581
	10,000	100		v3:	7.76 [8.00]%		6,738 [6,865]	
				v0, v2:	9.61 [10.00]%			
	0	100		v6:	3.96 [4.00]%	6.99 [7.20]%	3,402 [3,413]	2,601
	1000	100		v1, v3:	5.88 [6.00]%		3,752 [3,773]	
				v0, v2:	9.61 [10.00]%			

Figure 6.6: Low risk on route solution details without and with additive risk approximation

routing cost is the same for all combinations. Moreover, for 2% leg risk, the maximum vehicle risk can be observed as well for vehicle v6 (11.42% and respectively 12% on additive approximation) across all combinations, thereby exceeding the threshold of 10%. The solutions obtained with additive risk approximation have as well slightly higher cost but the same routes and routing cost.

Then, for 2% leg risk and a maximum vehicle risk of 10%, we can observe that the maximum vehicle risk is 9.61% and respectively 10% on additive approximation for different vehicles

across all combinations. We can see in this case that for the combinations $\langle 1000, 10 \rangle$ and $\langle 10000, 10 \rangle$, the solutions obtained with additive risk approximation have a slightly higher routing cost ($2581 > 2573$) and therefore different routes (corresponding to the solutions obtained for combinations $\langle 0, 10 \rangle$ and $\langle 10000, 100 \rangle$). Since we have different routes, the solutions obtained without risk approximation have been reassessed with risk approximation and vice-versa. We can note that for combination $\langle 1000, 10 \rangle$, the solution cost is the same: 2997. However, for combination $\langle 10000, 10 \rangle$, the solution cost initially obtained without risk approximation is 5948 (including 2573 routing cost). In contrast, a larger solution cost of 5954 (including 2581 routing cost) is obtained without using the risk approximation on the routes initially obtained with risk approximation. This hints that there is a limit on the usefulness on the additive risk approximation, even though for most combinations the routes obtained with additive risk approximation are the same as those obtained without. The approximation is useful up to about 2% risk per leg.

Figure 6.7 provides the solution cost analysis carried out by evaluating against one another the solutions depicted in Figure 6.5 for a leg risk of 1%. More specifically, each solution obtained for a particular $\langle \text{unit vehicle cost}, \text{unit cargo cost} \rangle$ combination was re-evaluated for every other $\langle \text{unit vehicle cost}, \text{unit cargo cost} \rangle$ combination corresponding to the other solutions. This was done in order to verify that each of the obtained solutions has the best cost for its corresponding

solCost	2402	2456	2905	2559	2613	3072	3967	4021	4506	unitVehCost	unitCargoCost
2402 <=	.	<u>2402</u>	2422	<u>2402</u>	<u>2402</u>	2422	<u>2402</u>	<u>2402</u>	<u>2402</u>	0	0
2456 <=	<u>2456</u>	.	2471	<u>2456</u>	<u>2456</u>	2471	<u>2456</u>	<u>2456</u>	<u>2456</u>	0	10
2905 <=	2941	2941	.	2941	2941	<u>2905</u>	2491	2491	2491	0	100
2559 <=	<u>2559</u>	<u>2559</u>	2589	.	<u>2559</u>	2589	<u>2559</u>	<u>2559</u>	<u>2559</u>	1000	0
2613 <=	<u>2613</u>	<u>2613</u>	2638	<u>2613</u>	.	2638	<u>2613</u>	<u>2613</u>	<u>2613</u>	1000	10
3072 <=	3098	3098	<u>3072</u>	3098	3098	.	3098	3098	3098	1000	100
3967 <=	<u>3967</u>	<u>3967</u>	4090	<u>3967</u>	<u>3967</u>	4090	.	<u>3967</u>	<u>3967</u>	10000	0
4021 <=	<u>4021</u>	<u>4021</u>	4139	<u>4021</u>	<u>4021</u>	4139	<u>4021</u>	.	<u>4021</u>	10000	10
4506 <=	<u>4506</u>	<u>4506</u>	4573	<u>4506</u>	<u>4506</u>	4573	<u>4506</u>	<u>4506</u>	.	10000	100
unitVehCost	0	0	0	1000	1000	1000	10000	10000	10000		
unitCargoCost	0	10	100	0	10	100	0	10	100		

Figure 6.7: Solution cost analysis for low risk of traversal

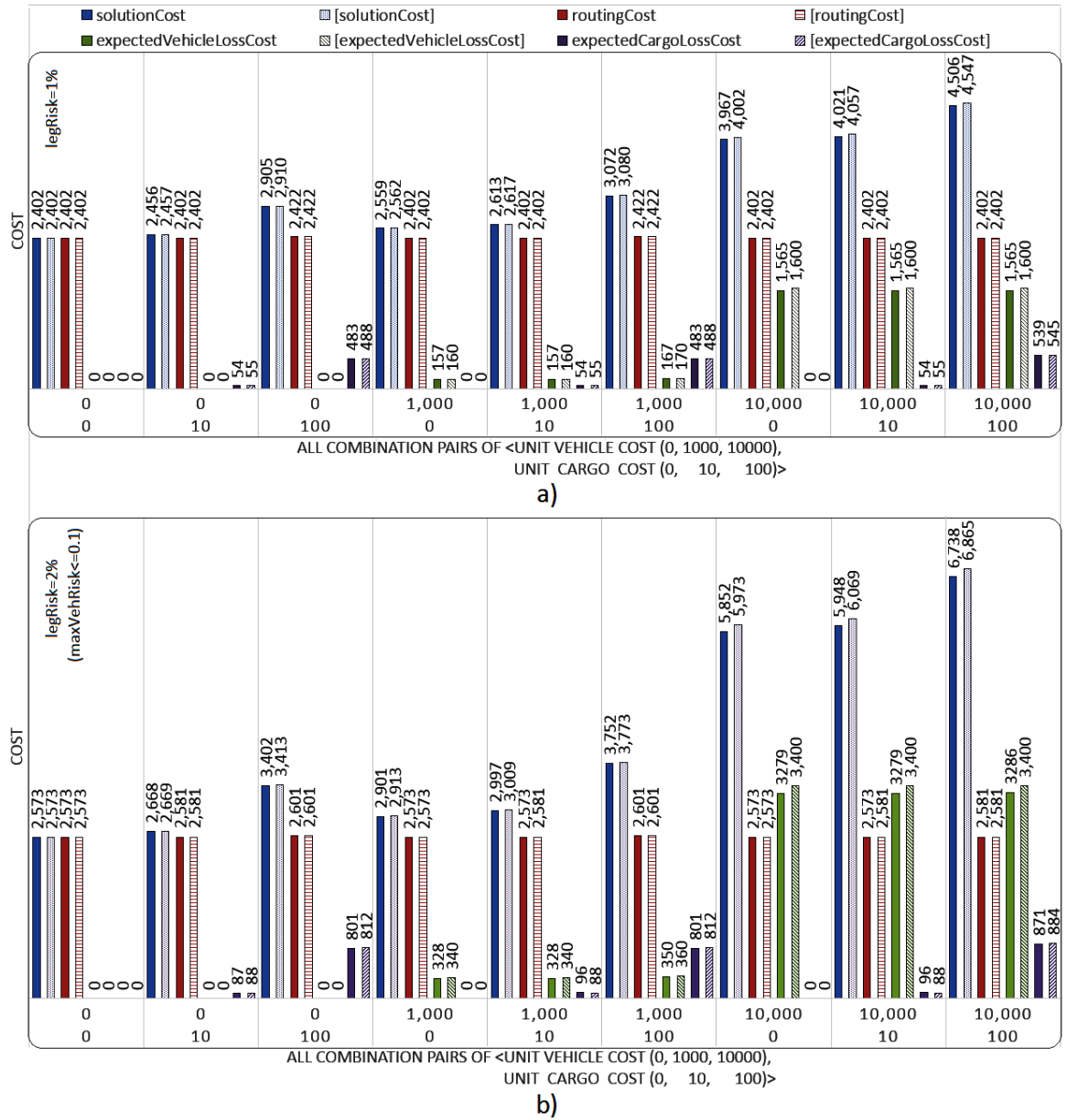


Figure 6.8: Solution sensitivity cost breakdown without and with (marked by [...]) risk approximation: 1% leg risk, solutions still under threshold and 2% leg risk, max. vehicle risk applied

<unit vehicle cost, unit cargo cost> combination. We note that since many solutions share the same routes, there are numerous cases where the same solution cost (underlined in the table) is obtained relative to the particular <unit vehicle cost, unit cargo cost> combination verified against.

Figure 6.8 depicts the solution cost breakdown and comparison for risk on route sensitivity (1% leg risk in Figure 6.8(a) and 2% leg risk in Figure 6.8(b)) relative to the routing cost, expected vehicle loss cost and expected cargo loss cost, without and with additive risk approximation. In the latter case, since the risk on route is slightly overestimated with each additional leg, we can note that the solutions have a slightly increased overall cost even though in most cases (except <1000, 10> and <10000, 10>), the routing cost is the same. We can also note that for the small leg risk values considered, the routing cost is dominant in most situations.

6.4.2 Route Generation on High Risk of Vehicle Failure

In the second part of the case study, we consider larger risk values where the linear approximation is not suitable and solve the problem using the same heuristic algorithm. Consequently, the actual cumulative risk value is assessed over each route.

We consider next a risk value of 10% for each vehicle movement across every edge of the problem. Then, an even larger risk value of 20% is also considered in order to illustrate the solution impact corresponding to higher risk values. Figure 6.9 depicts the 9 corresponding solutions (a,b,c,d,e,f,g,h and i) obtained for a vehicle risk of travel value of 10% on every edge of the problem. In particular, Figure 6.9(a) depicts the solution obtained when considering no impact of risk on route <unit vehicle cost=0, unit cargo cost=0>, which results in obtaining the solution of the original MDVRP problem. Then, Figure 6.9(b) shows the solution obtained for <unit vehicle cost=0, unit cargo cost=10>. We can note that compared to the original solution, it has an additional vehicle starting at depot D3. Figure 6.9(c) shows the solution obtained for <unit vehicle cost=0, unit cargo cost=100>. Compared to Figure 6.9(b), it has three additional vehicles (2 more at depot D1 and 1 more at depot D3). Figure 6.9(d) shows the solution obtained for <unit vehicle cost=1000, unit cargo cost=0>. We can note that it has the same number of vehicles (3) as the original solution, of which only one (v0) has the same route. Figure 6.9(e) shows the solution obtained for <unit vehicle cost=1000, unit cargo cost=10>. The routes seem to be those of the original solution but the visiting order of the demand nodes is different such

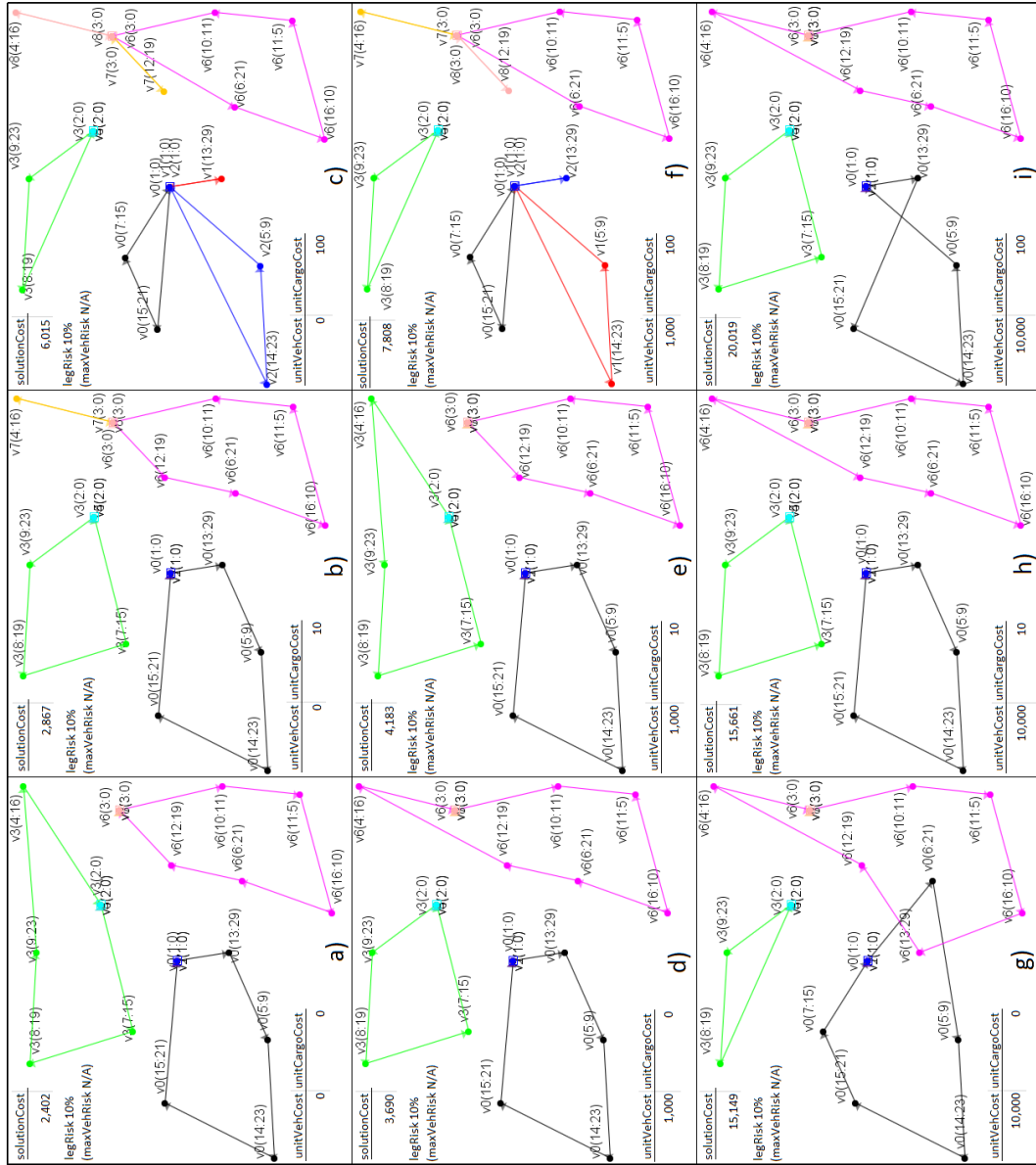


Figure 6.9: Risk on route solution sensitivity (10% leg risk, max. vehicle risk not applied)

that the vehicles mostly serve the larger demands before the lower ones (v3 and v6). Figure 6.9(f) shows the solution obtained for $\langle \text{unit vehicle cost}=1000, \text{unit cargo cost}=100 \rangle$. We can note that it has the same routes as in Figure 6.9(c) although some routes are interchanged between some vehicles of the same depots (v1 with v2 and v7 with v8). Figure 6.9(g) shows the solution obtained for $\langle \text{unit vehicle cost}=10000, \text{unit cargo cost}=0 \rangle$. We can note that it has 3 vehicles but the routes are different compared to Figure 6.9(a) and Figure 6.9(d). Figure 6.9(h) shows the solution obtained for $\langle \text{unit vehicle cost}=10000, \text{unit cargo cost}=10 \rangle$. Compared to Figure 6.9(d), the routes seem to be the same. However, the visiting order of the demand nodes for vehicle v6 is such that the larger demands are generally served before the smaller ones. Figure 6.9(i) shows the solution obtained for $\langle \text{unit vehicle cost}=10000, \text{unit cargo cost}=100 \rangle$. The routes are similar compared to those in Figure 6.9(h), with the same nodes assigned to the same vehicles. However, the demand node visiting order for v0 is such that the smallest demand is served last.

Figure 6.10 provides a spreadsheet containing the details for all solutions depicted in Figure 6.9. It also provides the details for the solutions obtained with a restriction of maximum vehicle risk on route of 50% over selected combinations of $\langle \text{unit vehicle cost}, \text{unit cargo cost} \rangle$. In this combinations the corresponding unrestricted solutions include vehicles with tour risks exceeding the threshold of 50%. In addition, the spreadsheet provides for the same selected combinations of $\langle \text{unit vehicle cost}, \text{unit cargo cost} \rangle$ the details for the corresponding solutions obtained with a higher risk value of 20% for vehicle leg movement on every edge (without and with a restricted maximum vehicle risk on route of 50%). The spreadsheet has 8 columns corresponding in the listed order to the following metadata: leg risk/(maximum vehicle risk); unit vehicle cost; unit cargo cost; solution vehicle count; vehicle risk (minimum values to maximum values); average vehicle risk; solution cost; routing cost. The first line of the spreadsheet corresponds to the aforementioned metadata. The next 9 lines provide the details for the 9 solutions depicted in Figure 6.9. We can note that in 4 out of the 9 solutions, there are vehicles with a corresponding tour risk exceeding 50%. More specifically, these 4 solutions correspond to the following $\langle \text{unit vehicle cost}, \text{unit cargo cost} \rangle$ combinations: $\langle 1000, 0 \rangle$; $\langle 10000, 0 \rangle$; $\langle 10000, 10 \rangle$; and $\langle 10000, 100 \rangle$.

Then, the next 4 lines provide the details for the solutions obtained with a restricted maximum vehicle risk on route of 50% over the aforementioned 4 combinations. Also, the corresponding routes for the combinations $\langle 10000, 0 \rangle$; $\langle 10000, 10 \rangle$; and $\langle 10000, 100 \rangle$ are shown in

legRisk	unitVehCost	unitCargoCost	solVehCnt	vehRisk (min to max)		averageVehRisk	solCost	routingCost
legRisk=10% (maxVehRisk N/A)	0	0	3	v0, v3:	40.95%	42.92%	2,402	2,402
				v6:	46.86%			
	0	10	4	v7:	19.00%	35.3%	2,867	2,422
				v3:	34.39%			
				v0:	40.95%			
				v6:	46.86%			
	0	100	7	v1, v7, v8:	19.00%	25.61%	6,015	2,974
				v0, v2, v3:	27.10%			
				v6:	40.95%			
	1,000	0	3	v3:	34.39%	42.5%	3,690	2,414
				v0:	40.95%			
				v6:	52.17%			
	1,000	10	3	v0, v3:	40.95%	42.92%	4,183	2,402
				v6:	46.86%			
	1,000	100	7	v2, v7, v8:	19.00%	25.61%	7,808	2,974
				v0, v1, v3:	27.10%			
				v6:	40.95%			
	10,000	0	3	v3:	27.10%	42.04%	15,149	2,536
				v0:	46.86%			
				v6:	52.17%			
	10,000	10	3	v3:	34.39%	42.5%	15,661	2,414
				v0:	40.95%			
				v6:	52.17%			
	10,000	100	3	v3:	34.39%	42.5%	20,019	2,495
				v0:	40.95%			
				v6:	52.17%			
legRisk=10% (maxVehRisk<=0.5)	1,000	0	3	v0, v3:	40.95%	42.92%	3691	2,402
				v6:	46.86%			
	10,000	0	3	v3:	34.39%	42.7%	15252	2,441
				v0, v1:	46.86%			
	10,000	10	3	v3:	34.39%	42.7%	15745	2,441
				v0, v1:	46.86%			
	10,000	100	3	v0, v3:	40.95%	42.92%	20091	2,483
				v6:	46.86%			
legRisk=20% (maxVehRisk N/A)	0	0	3	v0, v3:	67.23%	69.41%	2,402	2,402
				v6:	73.78%			
	1,000	0	3	v3:	59.04%	68.43%	4,466	2,414
				v0:	67.23%			
				v6:	79.02%			
	10,000	100	3	v0, v1:	59.04%	67.1%	31,603	2,745
legRisk=20% (maxVehRisk<=0.5)				v6:	83.22%			
	0	0	3	v8:	36.00%	46.97%	3,291	3,291
				v0-3, v6, v7:	48.80%			
	1,000	0	7	v8:	36.00%	46.97%	6,579	3,291
				v0-3, v6, v7:	48.80%			
	10,000	100	7	v8:	36.00%	46.97%	41,733	3,329
				v0-3, v6, v7:	48.80%			

Figure 6.10: High risk on route solution details

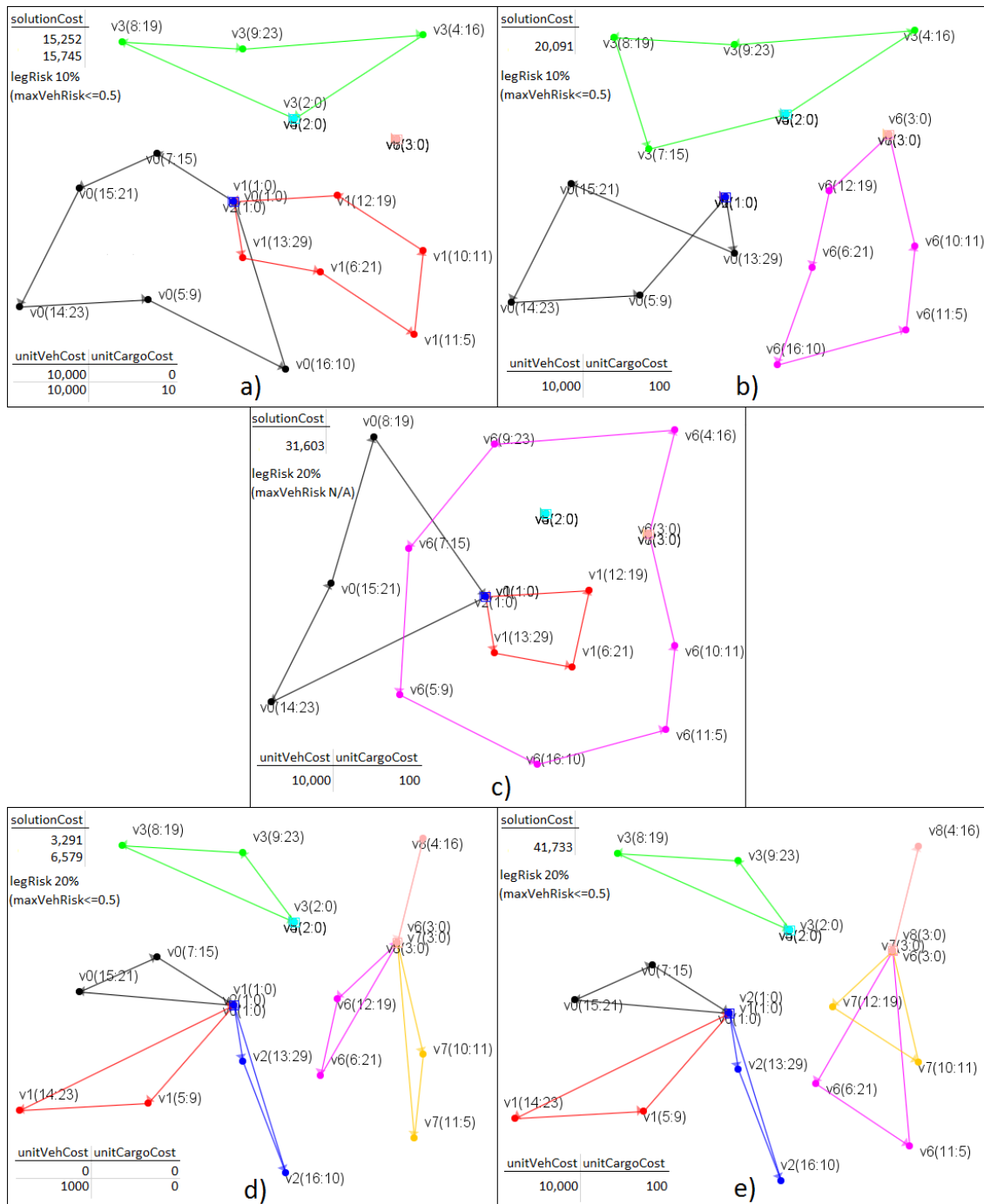


Figure 6.11: Risk on route solution sensitivity (10% leg risk, with max. vehicle risk; 20% leg risk, without and with max. vehicle risk)

solCost	2402	2867	6015	3690	4183	7808	15149	15661	20019	unitVehCost	unitCargoCost
2402 <=	.	2422	2974	2414	<u>2402</u>	2974	2536	2414	2495	0	0
2867 <=	2949	.	3277	2968	2894	3277	3133	2910	2972	0	10
6015 <=	7869	6867	.	7943	7318	<u>6015</u>	8506	7371	7268	0	100
3690 <=	3691	3835	4767	.	3691	4767	3798	<u>3690</u>	3771	1000	0
4183 <=	4283	4280	5070	4244	.	5070	4395	4186	4248	1000	10
7808 <=	9158	8280	<u>7808</u>	9219	8607	.	9768	8647	8544	1000	100
15149 <=	15278	16542	20899	15165	15278	20899	.	15165	15246	10000	0
15661 <=	15825	16987	21202	15719	15770	21202	15746	.	15723	10000	10
20019 <=	20745	20987	23940	20694	20194	23940	21119	20122	.	10000	100
unitVehCost	0	0	0	1000	1000	1000	10000	10000	10000		
unitCargoCost	0	10	100	0	10	100	0	10	100		

Figure 6.12: Solution cost analysis for high risk of traversal

Figure 6.11 (a) and (b). The last 6 lines correspond to 6 solutions (3 unrestricted and 3 restricted to maximum vehicle risk of 50%) obtained for a higher risk value of 20% on every vehicle leg, over the combinations $\langle 0, 0 \rangle$, $\langle 1000, 0 \rangle$ and respectively $\langle 10000, 100 \rangle$. The routes of the aforementioned solutions are shown in Figures 6.11(c), 6.11(d) and 6.11(e). We note that when there is no maximum vehicle risk applied, the solutions tend to use less vehicles, which are exposed to higher risk. We also note that when the per unit vehicle cost is higher (10000), there are situations where some depots start two vehicles while other depots start no vehicles (Figures 6.11(a) and 6.11(c)).

Figure 6.12 provides the solution cost analysis carried out by evaluating against one another the 9 solutions depicted in Figure 6.9. Thus, each solution for a particular $\langle \text{unit vehicle cost, unit cargo cost} \rangle$ combination was re-evaluated for every other $\langle \text{unit vehicle cost, unit cargo cost} \rangle$ combination corresponding to the other solutions. This was done to check that each obtained solution has the best cost for its $\langle \text{unit vehicle cost, unit cargo cost} \rangle$ combination. This is confirmed by the values shown in Figure 6.12. We note however, that there are 4 solutions that have the same cost for a second $\langle \text{unit vehicle cost, unit cargo cost} \rangle$ combination in addition to their respective one. This is due to the particular routes that may result in the same cost when the demand node visiting order is not relevant (unit cargo cost=0) or when the same routes result due to a high unit cargo cost of 100 coupled with a lower vehicle unit cost of 0 and respectively 1000.

Figures 6.13 and 6.14 depict the solution cost breakdown and comparison for risk on route sensitivity relative to the routing cost, expected vehicle loss cost and expected cargo loss cost. We can note that there are specific cases where either the routing cost, the expected vehicle loss cost or the expected cargo cost is dominant.

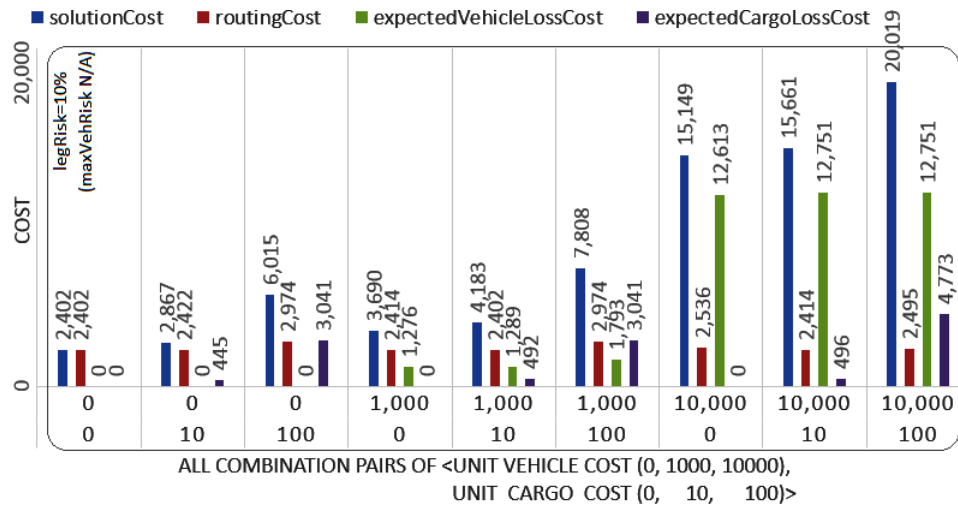


Figure 6.13: Solution cost breakdown and comparison for risk on route sensitivity

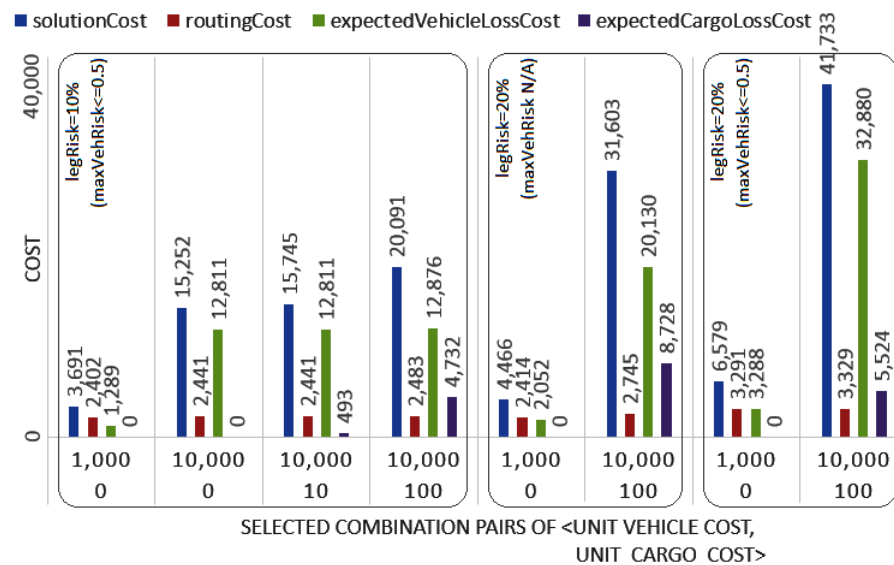


Figure 6.14: Solution cost breakdown and comparison for selected risk on route solutions

6.4.3 Route Replanning

We discuss next the application of replanning (adaptive planning) during logistic support plan execution, in the context of both proactive, reactive and mixed settings. We assume that the tasked vehicles are following their previously assigned routes up to the occurrence of an exogenous event that triggers the replanning process. We consider generic exogenous events that can change the risk exposure of the vehicles (e.g., weather fronts) as well as vehicle failure events.

In order to better convey the benefits of replanning, without loss of generality, we generate the replanning solution after the vehicles (except the ones that may have failed) complete their current legs, following a replanning triggering event. This allows to maintain the same problem data in terms of node locations where the vehicles are starting after replanning. Moreover, the routes assigned to the vehicles are marked with dotted lines while the current vehicle tracks are drawn over with solid lines up to the current position of the vehicle(s). Thus, Figure 6.15 presents 3 illustrative replanning scenarios exercised for a hypothetical extreme situation of crisis characterized by high risk on route (10% or more per leg traversal). In this setting, we use combination: $\langle \text{unit vehicle cost}=10000, \text{unit cargo cost}=100 \rangle$ along with maximum vehicle risk value of 50%.

The first scenario is depicted in Figure 6.15 (a) and Figure 6.15 (b) respectively. It involves a case of proactive replanning where due to an exogenous event, the edge traversal risk of the initial planning problem is changing from 10% to 20% during the execution of the plan. At the occurrence of the exogenous event, vehicles v0 (black track), v3 (green track) and v6 (magenta track) are traversing their first leg as shown in Figure 6.15 (a). The result of replanning can be seen in Figure 6.15 (b) where the routes of vehicles v0, v3 and v6 have been changed (shortened due to higher risk on route) while 4 supplementary vehicles with short routes have been employed as follows: 2 from depot 1, 1 from depot 2 and 1 from depot 3.

The second scenario is depicted in Figure 6.15 (c) and Figure 6.15 (d) respectively. It involves a case of reactive replanning in response to a vehicle failure event whereby vehicle v6 (magenta track) fails during the traversal of its second leg as shown in Figure 6.15 (c). The result of replanning can be seen in Figure 6.15 (d) where a new vehicle is employed from depot 1 in order to serve the demand nodes that can no longer be served by vehicle v6 from depot 3. We note that since node 12 was already served by v6, it becomes more cost effective to send vehicle v1 (red track) from depot 1 rather than another vehicle from depot 3.

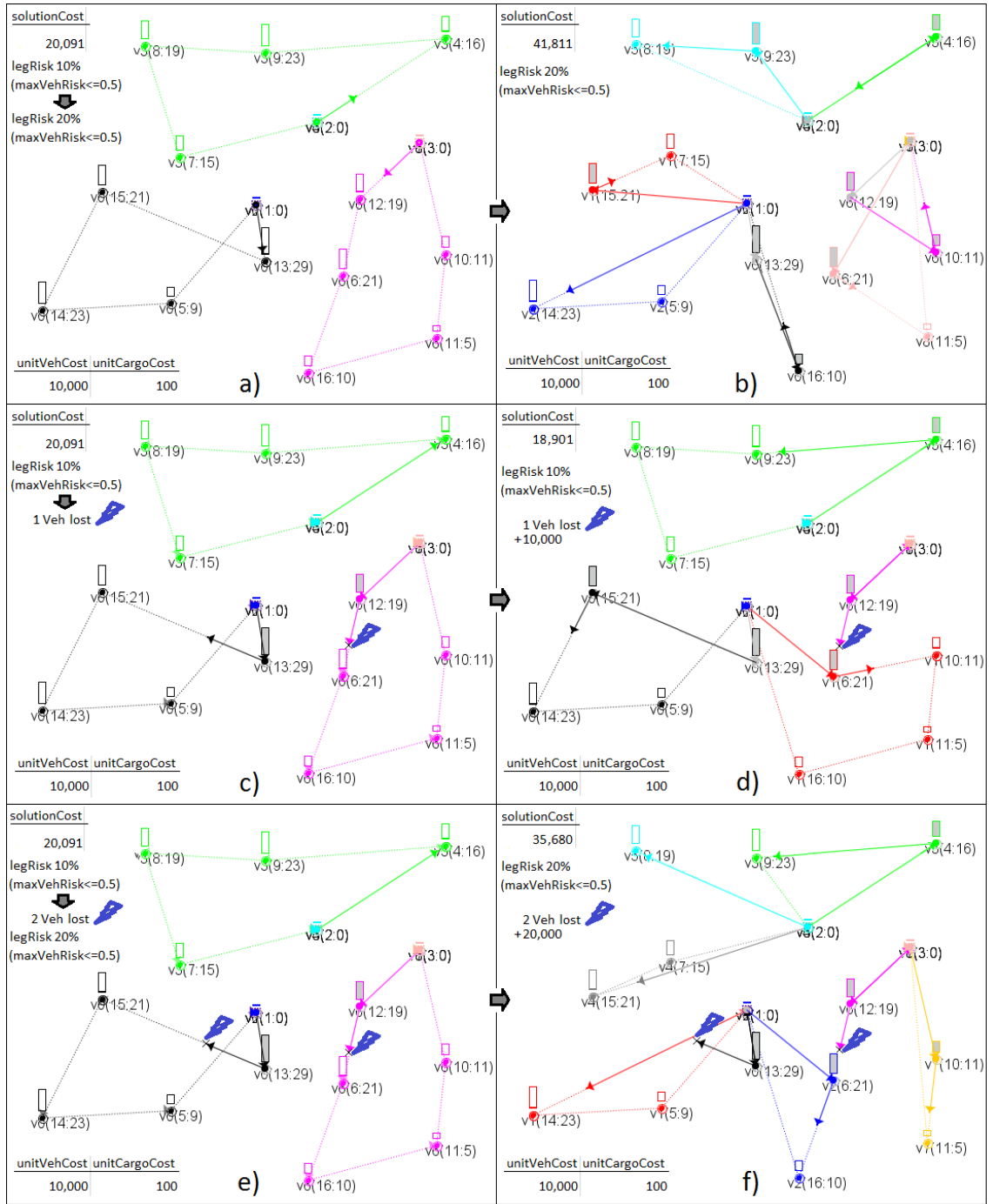


Figure 6.15: Proactive and reactive risk mitigation replanning

The third scenario is depicted in Figure 6.15 (e) and Figure 6.15 (f) respectively. It extends the second scenario and involves a case of reactive and proactive replanning in response to a vehicle failure event whereby two vehicles, namely v_0 (black track) and v_6 (magenta track) fail during the traversal of their second legs. This prompts as well the reconsideration of the edge traversal risk for the replanning problem from 10% to 20% as shown in Figure 6.15 (e). The result of the replanning solution can be seen in Figure 6.15 (f). The figure shows that the route of vehicle v_3 was changed (shortened due to higher risk on route) while 5 supplementary vehicles with short routes have been employed (2 from depot 1, 2 from depot 2 and 1 from depot 3). Compared to the second replanning scenario, we note that even though node 12 was served by v_6 from depot 3, due to considering 20% edge traversal risk, a new vehicle, v_7 (yellow track) is also sent in this case from depot 3 to serve two nodes not served by v_6 . In addition, the nodes not served by v_0 from depot 1, are in turn served from depot 1 by v_1 (red track) and from depot 2 by v_4 (gray track).

6.5 Benchmark Results

In order to assess the effectiveness of the proposed approach, a number of experiments have been performed over relevant problem instances. The instances actually represent risk augmented variants of original MDVRP instances from the P-series problem set assessed in [120]. The augmentation of the problems consists in associating vehicle risk traversal probabilities for all the edges of the underlying transport network. Specifically, for each problem we associate probabilities over the transport network edges, with uniformly distributed values between 1% and 2% risk of vehicle failure upon traversal.

We have selected the aforementioned risk of traversal values since they can capture reasonably well a hostile environment, as discussed in Section 6.2.2. Furthermore, the risk traversal values between any pair of nodes is symmetric. While not a limitation of the solution generation procedure, the use of symmetric edge traversal risk values allows to limit the potential impact on the routing cost portion of the combined routing and risk-related cost of the generated solutions. This helps to meaningfully compare the routing cost part of the risk constrained solutions against the best routing cost values of the corresponding original MDVRP instances assessed in [120]. In addition, we consider for most problem instances the value $maxRisk = 10\%$ for vehicle tour

maximum risk, which also represents a reasonable value for dispatching vehicles in situations of crisis. For only one instance (subsequently marked with *), the value $maxRisk = 13\%$ is considered due to limited vehicle availability relative to the number of demand nodes, which results in vehicle tours with a larger number of visited demand nodes per vehicle. Moreover, we use the value $PV = 1000$ for the per unit vehicle loss cost and respectively the value of $PD = 10$ for the per unit cargo loss cost. This $\langle PV = 1000, PD = 10 \rangle$ pair of low values has also been selected in order to limit the effect of potential vehicle and cargo loss on the heuristically generated solutions. This also helps to meaningfully compare the routing cost part of the RCMDVRP solutions against the best routing cost values of the corresponding MDVRP solutions.

Thus, the following P-series problem instances have been selected: p1, p2, \dots p7. These represent relevant size problems having between 50 to 100 demand nodes, and between 2 to 5 depots. Moreover, the number of vehicles per depot is ranging from 2 to 8 while the cargo capacities range from 80 to 200. Problem instance p5 in particular has 100 demand nodes, 2 depots and 5 vehicle per depot, which corresponds to an average of 10 demand nodes served per vehicle. Given that the transport network edge traversal risk is between 1% and 2%, this would result in a vehicle tour risk exposure greater than 10%. Thus, for p5 the value $maxRisk = 13\%$ is considered instead.

Table 6.1 presents the details of each problem along with the solution details. We note that problem instance p5 in particular has a star marking (p5*) to indicate that its solutions have a higher maximum vehicle risk. The solutions have been obtained with a Java implemented solver using the tuned search parameters available in [120]. For each problem instance, 8 solutions have been generated on a Core i7 platform with different randomizing seeds in order to provide minimum, maximum and average results.

Inst.	nodes	depot count	vehicles per depot	vehicle capacity	avg. no-risk routing only cost	avg. no-risk compute time(s)	min. heur. combined cost	max. heur. combined cost	avg. heur. combined cost	avg. heur. routing only cost	avg. heur. compute time(s)
p1	50	4	4	80	583.38	2.38	1656	1678	1670.13	637.13	3.18
p2	50	4	2	160	477.88	3.75	1651	1670	1660.25	622.38	13.75
p3	75	5	3	140	643.13	12.75	2429	2456	2443.00	822.88	13.88
p4	100	2	8	100	1007.13	52.13	3183	3205	3195.50	1217.50	68.51
p5*	100	2	5	200	758.63	41.00	3179	3247	3222.38	1176.75	456.75
p6	100	3	6	100	897.75	40.63	3050	3095	3075.25	1086.88	31.01
p7	100	4	4	100	897.00	25.13	3036	3094	3060.00	1074.63	26.63

Table 6.1: Benchmark results for RCMDVRP

The first column of Table 6.1 provides the instance name while the second column provides the number of nodes. The third column provides the number of depots. The fourth and fifth columns provide the number of vehicles per depot and respectively the capacity of each vehicle. Columns six and seven provide the average routing only cost and respectively the average computing time for the original MDVRP instances assessed by [120], where no-risk is considered. Columns eight and nine provide the minimum and respectively the maximum combined cost of the obtained risk constrained solutions. In addition, columns ten and eleven provide the average combined cost for the risk constrained solutions and respectively their average routing only cost. Finally, column twelve provides the average computing time for risk constrained solution generation. We can note that the computing time values are similar to those in column seven, except for problems p2 and p5 for which it is higher. This is due to the limited availability of vehicles for these problems, which entails a more extensive search of the solution space, especially for p5.

Table 6.2 provides the vehicle risk exposure details for each problem. We similarly note that problem instance p5 has a star marking (p5*) indicating higher maximum vehicle risk values. The first column provides the instance name. The second column provides the vehicle minimum risk across three sub-columns: lowest of the minimum vehicle risk values of all solutions; average of the minimum vehicle risk values for all solutions; highest of the minimum vehicle risk values of all solutions. The third column provides the average vehicle risk values for all solutions. The fourth column provides the vehicle maximum risk across three sub-columns: lowest of the maximum vehicle risk values of all solutions; average of the maximum vehicle risk values for all solutions; highest of the maximum vehicle risk values of all solutions. The fifth column provides the number of used vehicles across three sub-columns: lowest number of vehicles used of all solutions; average

Inst.	<u>vehMinRisk</u>			<u>vehAvgRisk</u>	<u>vehMaxRisk</u>			<u>vehUsed</u>		
	(lowest val. of all solutions)	(average val. for all solutions)	(highest val. of all solutions)	(average val. for all solutions)	(lowest val. of all solutions)	(average val. for all solutions)	(highest val. of all solutions)	(lowest val. of all solutions)	(average val. for all solutions)	(highest val. of all solutions)
p1	4.45%	4.99%	5.40%	7.08%	8.84%	9.16%	9.37%	11	11.125	12
p2	6.79%	7.93%	8.60%	9.07%	9.82%	9.94%	10.00%	8	8	8
p3	4.90%	5.81%	7.06%	8.26%	9.35%	9.73%	9.91%	13	13.375	14
p4	6.09%	7.18%	7.98%	8.92%	9.89%	9.92%	9.99%	16	16	16
p5*	11.13%	11.75%	12.13%	12.58%	12.79%	12.94%	13.00%	10	10	10
p6	4.55%	5.61%	6.70%	8.51%	9.72%	9.93%	10.00%	17	17.25	18
p7	6.57%	7.34%	8.09%	8.89%	9.72%	9.89%	9.98%	16	16	16

Table 6.2: Vehicle risk exposure for RCMDVRP solutions

number of vehicles used for all solutions; highest number of vehicles used of all solutions. We can see that for each problem instance, the maximum vehicle risk constraint is satisfied in each case, as indicated by the highest of the maximum vehicle risk values for the obtained solutions. As such, the maximum vehicle risk is under 10% for p1, p3, p4 and p7. Moreover, it is at most 10% for p2 and p6 while being at most 13% for p5.

Inst.	MDVRP solutions evaluated with risk on route					RCMDVRP obtained solutions		
	vehRisk			vehUsed	combined cost routing+vehLoss+cargoLoss	vehUsed	combined cost routing+vehLoss+cargoLoss	savings
	min	avg	max					
p1	3.66%	8.22%	10.19%	11	577+ 906+ 297=1780	11	638+ 776+242=1656	7.49%
p2	13.00%	15.45%	17.85%	5	472+ 773+ 572=1817	8	612+ 733+306=1651	10.05%
p3	8.71%	11.10%	14.61%	11	638+1221+ 751=2610	14	789+1126+514=2429	7.45%
p4	7.17%	11.13%	14.87%	15	997+1669+ 775=3441	16	1240+1403+540=3183	8.11%
p5*	14.01%	18.83%	22.95%	8	749+1505+1362=3616	10	1146+1257+776=3179	13.75%
p6	6.93%	10.49%	15.09%	16	890+1680+ 696=3266	17	1082+1453+515=3050	7.08%
p7	3.22%	10.28%	13.89%	16	883+1645+ 742=3270	16	1073+1408+555=3036	7.71%

Table 6.3: Best MDVRP solution evaluation in presence of risk vs. the savings achieved by the best RCMDVRP solutions

Considering risk on route, Table 6.3 shows the results of evaluating vehicle risk exposure and the combined routing and risk-related cost corresponding to the best solutions of the original MDVRP instances assessed in [120]. This allows for an appraisal of the benefits offered by the obtained risk constrained solutions in terms of risk mitigation and cost. As such, the table indicates as well the cost savings achieved by the best RCMDVRP solutions for the same problem instances.

In order to make the comparison more challenging, all vehicle tours of each evaluated MDVRP solution have been assessed in both clockwise and counter-clockwise manner in pursuit of the lowest combined routing and risk-related cost (routing cost + potential vehicle loss cost + potential cargo loss cost). This was possible since the routing cost and the traversal risk probability values are symmetric across the transport network edges.

The first column of Table 6.3 indicates the instance name. The second, third and fourth columns provide respectively the minimum, average and maximum vehicle risk values corresponding to the tours of the evaluated MDVRP solutions. The fifth column provides the number of used vehicles in the evaluated MDVRP solutions. The sixth column provides the combined routing and risk-related cost for the evaluated MDVRP solutions. The seventh column provides the number of used vehicles in the RCMDVRP solutions. The eighth column provide the combined routing and

risk-related cost for the RCMDVRP solutions. Finally, the ninth column provides the savings, in terms of corresponding percentage, achieved by the RCMDVRP solutions relative to the combined cost of the evaluated MDVRP solutions.

We can note that all evaluated MDVRP solutions in the presence of risk on route, exhibit higher maximum vehicle risk for some of their tours, compared to the threshold value allowed for their RCMDVRP counterparts. As such, the maximum vehicle risk exceeds 10% for p1, p2, p3, p4, p6 and p7, ranging from 10.19% to 17.85%. Likewise, the maximum vehicle risk for p5 is 22.95%, thereby notably exceeding 13%. We also note that for p2 and p5, all vehicles have higher risk than maximum threshold since the minimum vehicle risk is 13.00 for p2 and 14.01 for p5. Furthermore, compared to the RCMDVRP solutions, the number of vehicles used in the evaluated MDVRP solutions is generally smaller (except for p1 and p7 where it is the same), and the routing cost portion of the combined cost is also smaller. However, compared to the RCMDVRP solutions, the combined routing and risk-related cost is higher (due to comparatively higher potential vehicle and cargo loss). In contrast, the corresponding RCMDVRP solutions exhibit cost savings ranging from 7.08% to 13.75% (due to comparatively lower potential vehicle and cargo loss).

6.6 Summary

This chapter discussed the need for adaptive logistic planning with risk mitigation to address the challenges of plan execution in perilous environments where the logistic component of a collaborative plan can be impacted by exogenous events and hostile activities. Such capability is instrumental for a planning/replanning decision support system, which can take into account logistic delivery risk mitigation and proactive/reactive plan adaptation in response to changes in risk factors and exogenous events leading to vehicle breakdown or loss. In this pursuit, the underlying risk constrained multi depot vehicle routing problem (RCMDVRP) was approached by proposing a linear-approximation model as well as a heuristic solution generation technique.

The contribution consists in the elaboration of a model along with a learning-based heuristic technique that allows to minimize the combined vehicle routing cost, potential vehicle breakdown/loss cost and the potential cost of commodity delivery failure. The solution search mechanism employs a seed-based pseudo-random number generator, which steers the course of solution

search. This allows for parallel/decentralized solution generation across different computing nodes that can belong to specific decision making participants, typically located at depots. This can enable a team of decision making participants to collaboratively solve a common instance of a risk constrained logistic support planning/replanning problem. Moreover, an illustrative case study on planning/replanning activities was presented in both low and high risk settings. The case study highlighted the impact on solution generation when considering different values for vehicle risk on route along with various combinations of per unit vehicle loss cost and per unit cargo loss cost. The plan adaptation capability offered by the heuristic procedure is noteworthy especially in situations involving logistics deployment in uncertain environments such as those encountered during disaster relief, humanitarian aid distribution and rescue operations. Also, obtaining cost effective planning/replanning solutions that limit the potential cost that may be incurred due to vehicle risk exposure, represents another area of major interest for organisational decision makers.

In contrast to [140] and [141], which employ constructive heuristics and respectively ant colony metaheuristic for the risk-constrained cash-in-transit VRP, where no cost is considered for vehicle loss, the approach in this chapter handles multi-depot VRP with risk mitigation. In this setting, both vehicle and cargo loss are considered while employing a learning based solution generation heuristic. Compared to [46], which employs fuzzy simulation-based heuristics for minimizing total expected transport risk when delivering hazardous materials from multiple depots, the approach in this chapter minimizes the combined routing and potential vehicle and cargo loss cost, under a specified maximum risk threshold.

The solution generation approach provides competitive solutions that result from combining a learning guided stochastic cost insertion gradient descent technique with iterative solution improvement. Concerning the limitations, given that the approach employs a learning guided generative solution technique, a typical challenge relates to the generation of increasingly competitive solution fragments over the vast combinatorial solution search spaces characterizing larger problems. In such cases, the guided learning can have a limited effectiveness. However, the proposed approach has good performance for practical size problems. In this context, the heuristic technique was evaluated via benchmark results over a data set with problems ranging from 50 to 100 nodes. The results that have been obtained indicate that the approach can provide in short time cost effective solutions for risk constrained logistic delivery problems of relevant size.

Chapter 7

Conclusion

In the context of collaborative organizational planning, plan execution monitoring and plan adaptation in partially known and potentially hostile environments, participants seek advanced decision support capabilities. These include the assessment of plan associated risk, contingency options evaluation, appropriate shared information awareness according to different hierarchy levels, asset localization as well as adaptive planning during plan execution. Such capabilities are especially important when the plan execution takes place in the presence of communication disruption and exogenous events, which may impact ongoing tasks, potentially requiring plan adaptation. In this setting, the present thesis pursued four research directions relevant for the aforementioned decision support capabilities. In Chapter 1, the corresponding problems have been introduced in the context of providing logistic support over risk-prone transport networks since many organizational plans have important logistic support components. Then, Chapter 2 further discussed this problems and presented an overview of related research initiatives previously undertaken.

In terms of contributions, Chapter 3 introduced an approach that leverages probabilistic model-checking to assess risk-related properties for plans involving transportation tasks over different available routing options with various degrees of uncertainty. The proposed approach allows evaluating risk exposure, cost bounds, and contingency options for risk mitigation. The approach also allows the construction of decision trees from the model-checking results, which facilitate decision making by offering an enhanced and more informative risk assessment contrasting different outcomes of interest. In this respect, noteworthy areas of applicability relate to logistic support planning under uncertainty as encountered in humanitarian aid provision and rescue operations.

The next contribution in Chapter 4 presented a distributed and hierarchy-aware information sharing protocol suitable for achieving high level of shared information awareness in execution environments with noisy communication. The proposed approach makes use of a sliding window for information dissemination along with an asymmetric clustering technique whereby the participants observe organizational hierarchy. In this way, the information is aggregated at the top levels from the levels below. The information sharing mechanism has been assessed via probabilistic model-checking for different cluster arrangements of plan executing participants and various communication disruption levels. A typical area of applicability is related to supporting plan execution monitoring and adaptive planning activities during collaborative plan execution in hostile environments prone to erroneous communication.

Thereafter, Chapter 5 contributed an efficient management approach for resource constrained sensor networks in pursuit of asset localization during plan execution. The proposed approach allows to effectively employ resource constrained sensor networks in order to monitor the location of plan executing (or potentially plan interfering) assets. The elaborated heuristic technique allows to effectively manage the allocation of limited sensor network resources in terms of sensor range, capacity and energy use. This allows to minimize the overall localization error of the covered target assets, which can greatly benefit adaptive planning.

The last contribution was provided by Chapter 6, which proposed a cost effective approach for logistic planning and replanning with risk mitigation. The underlying heuristic technique can be very useful for organizations executing autonomous or collaborative plans involving logistic support in risk-prone environments. In this context, planning and replanning represent critical aspects for minimizing the cost of potential vehicle breakdown/loss and cargo delivery failure. The approach offers decision makers the possibility to both proactively adapt their plans with respect to evolving levels of perceived risk and to quickly replan in case of failed or lost vehicles.

Instructive case studies have also been documented for each of the proposed approaches along with benchmark results that have been generated for the proposed heuristic techniques in order to evaluate their effectiveness.

In terms of future work, concerning the first contribution, a potentially relevant research avenue that can be explored is in the context of logistic support monitoring, related to factoring evolving plan parameters that may affect potential outcomes during plan execution. As such, a

particular route may be initially selected after assessing it as being less risky and having a cost-wise beneficial outcome. However, the assessment result may change in response to evolving plan parameters such as restricted vehicle availability for replanning. In such case, the stakeholders could be provided with a suggestion for another route with a better potential outcome, if available.

With respect to the second contribution, a future direction consists in endowing plan executing participants with the ability to approximately infer their level of shared information awareness during plan execution. This can result from analysing the patterns of previously generated data in conjunction with the received data from other participants. The communication of invalid information (e.g., due to malfunctioning hardware) represents another future work avenue whereby the participants may decide to not disseminate certain event data should they recognize it as invalid.

Regarding the third contribution, an interesting further research direction to pursue consists in extending the sensor management procedure in order to improve potential target coverage by determining sensor locations as needed. Another interesting direction relates to selecting the type of sensors to be deployed for a given budget, factoring the acquisition cost of the sensors where more capable sensors have better coverage and target focusing capacity but higher cost.

For the fourth contribution, a future research avenue worthy to explore is to employ a heterogeneous vehicle fleet for adaptive logistic support with risk mitigation. In this respect, one may consider that for particular conditions specific for different segments of road or terrain, certain types of vehicle may be exposed to higher or lower risk due to features such as size, speed, etc. Another direction relates to considering risk for depots, which may affect vehicle availability.

In terms of gained insights, it is important to emphasize the benefits provided by the use of decision trees, the adoption of formal assessment techniques based on probabilistic model-checking as well as the employment of learning-based heuristics for difficult combinatorial optimization problems. Thus, decision trees can be instrumental for decision makers faced with difficult to compare potential outcomes of interest, during planning and replanning activities. However, decision trees have specific limitations such as being susceptible to small changes in the input data, which may result in a significant change in outcomes. Probabilistic model-checking represents a powerful technology, with the potential to automate the comprehensive assessment of probabilistic properties capturing risk exposure as well as communication under disruption, as encountered during the planning and plan execution processes. This allows to cover particular corner cases that

might be missed by non-comprehensive assessment techniques such as those based on simulation. Notwithstanding, model-checking has inherent limitations due to the problem of state explosion, which precludes large scale applications. Concerning the employment of learning-based heuristics, they are essential to quickly obtain effective solutions for sensor network management and vehicle routing during the planning and adaptive planning of logistic support in potentially hostile environments. The learning aspect of such heuristic procedures allows for an informed exploration of the humongous solution spaces of the combinatorial problems that they are aimed for. Consequently they scale well, allowing to efficiently compute near-optimal solutions in a short time, albeit exhibiting a specific solution quality degradation with the increase in the problem size.

Combined, the findings stemming from the four research directions have the potential to enable the elaboration of an aggregated framework for enhanced advisory decision support. Such framework would be suitable for both off-line and on-line assessment of the logistic support components of organizational plans. It would also allow for adaptive replanning with risk mitigation in response to changes detected in the execution environment, leveraging hierarchy-aware shared information awareness and asset localization during plan execution. However, a fully-fledged decision support framework for logistic planning and plan execution monitoring in partially known and potentially hostile environments requires both the employment of appropriate solution generation techniques as well as addressing specific technical aspects. These relate to software libraries, data storage and access, etc. The aforementioned aspects have associated complexity levels stemming from issues such as software library compatibility, data format or framework usage policy.

In relation to employing appropriate solution generation techniques, the present thesis, attempts to partially bridge the gap, within the scope of the carried out research. In this respect, the adoption and integration of the elaborated approaches can definitely benefit such a fully-fledged decision support framework. Moreover, one can anticipate a growing potential for organizational collaboration and information sharing stemming from an increased availability of cost effective data connectivity and computing power. This will further promote the adoption of collaborative platforms for logistic planning, plan execution monitoring and plan adaptation in perilous environments. Thus, a collaborative decision support platform can be engineered in future, by including the proposed techniques for logistic planning risk assessment, hierarchy-aware shared information awareness, asset localization and adaptive planning with risk mitigation.

Bibliography

- [1] Robert Abo and Kamel Barkaoui. A performability analysis of mobile wireless sensor networks with probabilistic model checking. In *Wireless Advanced (WiAd), 2011*, pages 283–288. IEEE, 2011.
- [2] Erika Abrahám, Nils Jansen, Ralf Wimmer, J Katoen, and Bernd Becker. DTMC model checking by SCC reduction. In *Quantitative Evaluation of Systems (QEST), 2010 Seventh International Conference on the*, pages 37–46. IEEE, 2010.
- [3] Badr Afify, Sujoy Ray, Andrei Soeanu, Anjali Awasthi, Mourad Debbabi, and Mohamad Allouche. Evolutionary learning algorithm for reliable facility location under disruption. *Expert Systems with Applications*, 115:223–244, 2019.
- [4] Nabeel Ahmed, David Hadaller, and Srinivasan Keshav. Incremental maintenance of global aggregates. Tech. Report CS-2006-19, 2006.
- [5] Enrique Alba and Bernabé Dorronsoro. Computing nine new best-so-far solutions for capacitated VRP with a cellular genetic algorithm. *Information Processing Letters*, 98(6):225–230, 2006.
- [6] Mohamed Ziad Albari. A taxonomy of runtime software monitoring systems. http://www.informatik.uni-kiel.de/~wg/Lehre/Seminar-SS05/Mohamed_Ziad_Albari/vortrag.pdf, 2005. Last accessed on 1 March 2019.
- [7] Alessandro Aldini. Saving privacy in trust-based user-centric distributed systems. *SECUREWARE Journal*, pages 76–81, 2014.

- [8] James P. Allen, Kevin P. Barry, John M. McCormick, and Ross A. Paul. Plan execution monitoring with distributed intelligent agents for battle command. In *Battlespace Digitization and Network-Centric Systems IV*, volume 5441, pages 69–78. International Society for Optics and Photonics, 2004.
- [9] James P. Allen and John M. McCormick. Adaptive plan monitoring systems for military decision support. <http://www.aaai.org/Papers/Symposia/Spring/2005/SS-05-02/SS05-02-002.pdf>, 2005. Last accessed on 1 March 2019.
- [10] Youngwon Kim An, Seong-Moo Yoo, Changhyuk An, and B. Earl Wells. Rule-based multiple-target tracking in acoustic wireless sensor networks. *Computer Communications*, 51:81–94, 2014.
- [11] P. Anussornnitarn, S. Nof, and O. Etzion. Decentralized control of cooperative and autonomous agents for solving the distributed resource allocation problem. *International Journal of Production Economics*, pages 114–128, 2005.
- [12] Gürdal Arslan, Jason R. Marden, and Jeff S. Shamma. Autonomous vehicle-target assignment: A game-theoretical formulation. *Journal of Dynamic Systems, Measurement, and Control*, 129(5):584–596, 2007.
- [13] P. Augerat, J.M. Belenguer, E. Benavent, A. Corberan, D. Naddef, and G. Rinaldi. Computational results with a branch and cut code for the capacitated vehicle routing problem. Technical report, Universite Joseph Fourier, Grenoble, France, 1995.
- [14] Chen Avin and Carlos Brito. Efficient and robust query processing in dynamic environments using random walk techniques. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, pages 277–286. ACM, 2004.
- [15] A. Awasthi, S. Chauhan, Y. Lechevallier, M. Parent, and M. Proth. A data mining approach for adaptive path planning on large road networks. *Foundations of Computational Intelligence: Data mining, Special Issue, Springer Publications*, pages 297–320, 2009.

- [16] Zeinab Azarmand and Ensiyeh Neishabouri. Location allocation problem. In Reza Zanjirani Farahani and Masoud Hekmatfar, editors, *Facility Location*, pages 93–109. Physica-Verlag HD, 2009.
- [17] Adnan Aziz, Kumud Sanwal, Vigyan Singhal, and Robert Brayton. Verifying continuous time Markov chains. In *Proceedings of 8th International Conference on Computer Aided Verification (CAV'96)*, volume 1102, pages 269–276. Springer, 1996.
- [18] Christel Baier. *On algorithmic verification methods for probabilistic systems*. PhD thesis, University of Mannheim, 1998.
- [19] D. Balakrishnan and A. Nayak. An efficient approach for mobile asset tracking using contexts. *IEEE Transactions on Parallel and Distributed Systems*, 23(2):211–218, 2012.
- [20] Hans-Jürgen Bandelt, Arjan Maas, and Frits C. R. Spieksma. Local search heuristics for multi-index assignment problems with decomposable costs. *Journal of the Operational Research Society*, 55(7):694–704, 2004.
- [21] Benoît Barbot, Taolue Chen, Tingting Han, Joost-Pieter Katoen, and Alexandru Mereacre. Efficient CTMC model checking of linear real-time objectives. In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 128–142. Springer, 2011.
- [22] Aharon Ben-Tal, Byung Do Chung, Supreet Reddy Mandala, and Tao Yao. Robust optimization for emergency logistics planning: Risk mitigation in humanitarian relief supply chains. *Transportation research part B: methodological*, 45(8):1177–1189, 2011.
- [23] P. Berger, A. Gerstenfeld, and A. Zeng. How many suppliers are best? A decision-analysis approach. *Management Science (Omega)*, 32(1):9–15, 2004.
- [24] F. Besnard. On stochastic dynamic programming and its application to maintenance. In *Master Thesis*. Royal Institute of Technology, KTH School of Electrical Engineering, 2007.
- [25] L. Bianco, M. Caramia, S. Giordani, and V. Piccialli. Operations research models for global route planning in hazardous material transportation. *Handbook of OR/MS models in hazardous materials transportation, Operations Research and Management Science*, 193:49–101, 2013.

- [26] Igor Bisio, Andrea Sciarrone, and Sandro Zappatore. A new asset tracking architecture integrating RFID, Bluetooth Low Energy tags and ad hoc smartphone applications. *Pervasive and Mobile Computing*, 31:79–93, 2016.
- [27] Raouf Boutaba, Nadjib Achir, Marc St-Hilaire, and Eduardo Freire Nakamura. Planning and deployment of wireless sensor networks. *International Journal of Distributed Sensor Networks*, 10(1):198139, 2014.
- [28] Kris Braekers, Katrien Ramaekers, and Inneke Van Nieuwenhuyse. The vehicle routing problem: State of the art classification and review. *Computers & Industrial Engineering*, 99:300–313, 2016.
- [29] Rainer Burkard, Mauro Dell’Amico, and Silvano Martello. *Assignment problems: revised reprint*. Society for Industrial and Applied Mathematics (SIAM), 2012.
- [30] Jonathan Caulkins, Eric D. Hough, Nancy R. Mead, and Hassan Osman. Optimizing investments in security countermeasures: A practical tool for fixed budgets. *IEEE Security & Privacy*, 5(5):57–60, 2007.
- [31] Geiser Chalco Chalco, Marco Aurélio Gerosa, Ig Ibert Bittencourt, and Seiji Isotani. Automated instructional design for CSCL: A hierarchical task network planning approach. *Expert Systems with Applications*, 41(8):3777–3798, 2014.
- [32] Mahesh Chand, Tilak Raj, and Ravi Shankar. A comparative study of multi criteria decision making approaches for risks assessment in supply chain. *International Journal of Business Information Systems*, 18(1):67–84, 2015.
- [33] Martin Christopher and Hau Lee. Mitigating supply chain risk through improved confidence. *International Journal of Physical Distribution & Logistics Management*, 34(5):388–396, 2004.
- [34] Alessandro Cimatti, Edmund Clarke, Enrico Giunchiglia, Fausto Giunchiglia, Marco Pistore, Marco Roveri, Roberto Sebastiani, and Armando Tacchella. Nusmv 2: An open-source tool for symbolic model checking. In *Computer Aided Verification*, pages 359–364. Springer, 2002.

- [35] Koen Claessen. Safety property verification of cyclic synchronous circuits. *Electronic Notes in Theoretical Computer Science*, 88:55–69, 2004.
- [36] Robert T. Clemen and Terence Reilly. *Making Hard Decisions with DecisionTools Suite*. Duxbury, CA, 2001.
- [37] Jean-François Cordeau, Michel Gendreau, and Gilbert Laporte. A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks*, 30(2):105–119, 1997.
- [38] Graham Cormode, S. Muthukrishnan, and Ke Yi. Algorithms for distributed functional monitoring. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1076–1085. Society for Industrial and Applied Mathematics, 2008.
- [39] Jorge Cortes, Sonia Martinez, Timur Karatas, and Francesco Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on robotics and Automation*, 20(2):243–255, 2004.
- [40] B. Cukic, H. Ammar, and K. Leteef. Identifying high-risk scenarios of complex systems using input domain partitioning. In *Proceedings of 9th Int. Symposium on Software Reliability Engineering*. ISSRE, 1998.
- [41] Mads Dam and Rolf Stadler. A generic protocol for network state aggregation. In *Radionetenskap och Kommunikation (RVK)*, pages 14–16, 2005.
- [42] Jeffrey Dean and Sanjay Ghemawat. MapReduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.
- [43] Léa A. Deleris and Feryal Erhun. Risk management in supply networks using Monte-carlo simulation. In *Proceedings of the 37th Winter Simulation Conference*, pages 1643–1649, 2005.
- [44] N. Delgado, A. Q. Gates, and S. Roach. A taxonomy and catalog of runtime software-fault monitoring tools. *IEEE Transactions on Software Engineering*, 30(12):859–872, 2004.
- [45] Ibrahim Dogan. Analysis of facility location model using Bayesian networks. *Expert Systems with Applications*, 39(1):1092–1104, 2012.

- [46] Jiaoman Du, Xiang Li, Lean Yu, Ralescu Dan, and Jiandong Zhou. Multi-depot vehicle routing problem for hazardous materials transportation: A fuzzy bilevel programming. *Information Sciences*, 399:201–218, 2017.
- [47] N. Dulac. Dynamic safety and risk management in complex socio-technical systems. In *MIT-Albany SD Colloquium*. MIT, 2005.
- [48] T. Eiter, E. Erdem, and W. Faber. Plan reversals for recovery in execution monitoring. In *10th International Workshop on NonMonotonic Reasoning*, 2004.
- [49] Erhan Erkut and Armann Ingolfsson. Transport risk models for hazardous materials: revisited. *Operations Research Letters*, 33(1):81–89, 2005.
- [50] Ernst Moritz Hahn and Tingting Han and Lijun Zhang. Synthesis for PCTL in parametric Markov decision processes. In *Proceedings of 3rd NASA Formal Methods Symposium (NFM’11)*, volume 6617 of *LNCS*. Springer, 2011.
- [51] Reza Zanjirani Farahani, Masoud Hekmatfar, Behnam Fahimnia, and Narges Kazemzadeh. Hierarchical facility location problem: Models, classifications, techniques, and applications. *Computers & Industrial Engineering*, 68:104–117, 2014.
- [52] Klein S. Feder T., Hell P. and Motwani R. Complexity of graph partition problems. In *Symposium on the Theory of Computing*, 1999.
- [53] Ansgar Fehnker and Peng Gao. Formal verification and simulation for performance analysis for probabilistic broadcast protocols. In *Ad-Hoc, Mobile, and Wireless Networks, 5th International Conference, ADHOC-NOW 2006, Ottawa, Canada, August 17-19, 2006, Proceedings*, pages 128–141, 2006.
- [54] Beatriz Fernández-Muñiz, José Manuel Montes-Peón, and Camilo José Vázquez-Ordás. Occupational risk management under the OHSAS 18001 standard: analysis of perceptions and attitudes of certified firms. *Journal of Cleaner Production*, 24:36–47, 2012.
- [55] D. Gade and E. A. Pohl. Sample average approximation applied to the capacitated-facilities location problem with unreliable facilities. *Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 223(4):259–269, 2009.

- [56] Wu Gang, Zhang Huxing, Qiu Meikang, Ming Zhong, Li Jiayin, and Qin Xiao. A decentralized approach for mining event correlations in distributed system monitoring. *Journal of Parallel and Distributed Computing*, 73(3):330–340, 2013.
- [57] J. A. García, J. Fdez-Valdivia, F. J. Cortijo, and R. Molina. A dynamic approach for clustering data. *Signal Processing*, 44(2):181–196, 1995.
- [58] A. Gerevini and I. Serina. Fast plan adaptation through planning graphs: Local and systematic search techniques. In *Fifth International Conference on Artificial Intelligence Planning Systems*, pages 112–121, 2000.
- [59] Bruce Golden, S. Raghavan, and Edward A. Wasil. *The vehicle routing problem : latest advances and new challenges*. Springer, 2008.
- [60] Dries Goossens, Sergey Polyakovskiy, Frits C. R. Spieksma, and Gerhard J. Woeginger. The focus of attention problem. *Algorithmica*, 74(2):559–573, 2016.
- [61] Peng Guo, Wenming Cheng, and Yi Wang. Hybrid evolutionary algorithm with extreme machine learning fitness function evaluation for two-stage capacitated facility location problems. *Expert Systems with Applications*, 71:57–68, 2017.
- [62] Fredrik Gustafsson and Fredrik Gunnarsson. Localization in sensor networks based on log range observations. In *International Conference on Information Fusion*, pages 1–8. IEEE, 2007.
- [63] Ernst Moritz Hahn, Holger Hermanns, and Lijun Zhang. Probabilistic reachability for parametric Markov models. In *International Journal on Software Tools for Technology Transfer*, pages 3–19. Springer, 2011.
- [64] William Eugene Hart, Natalio Krasnogor, and James E. Smith. Memetic evolutionary algorithms. In *Recent advances in memetic algorithms*, pages 3–27. Springer, 2005.
- [65] Andrew Hinton, Marta Z. Kwiatkowska, Gethin Norman, and David Parker. PRISM: A tool for automatic verification of probabilistic systems. In *Tools and Algorithms for the Construction and Analysis of Systems, 12th International Conference, TACAS 2006 Held as*

- Part of the Joint European Conferences on Theory and Practice of Software, 2006, Vienna*, pages 441–444, 2006.
- [66] William Ho, George T. S. Ho, Ping Ji, and Henry C. W. Lau. A hybrid genetic algorithm for the multi-depot vehicle routing problem. *Engineering Applications of Artificial Intelligence*, 21(4):548–557, 2008.
 - [67] G. Hoa, C. Leea, H. Lau, and A. Ip. A hybrid intelligent system to enhance logistics workflow: An olap-based ga approach. *International Journal of Computer Integrated Manufacturing*, pages 69 – 78, 2006.
 - [68] Ling Huang, Minos Garofalakis, Anthony D. Joseph, and Nina Taft. Approximate decision making in large-scale distributed systems. In *NIPS Workshop: Statistical Learning Techniques for Solving Systems Problems (MLSys)*. Vancouver, BC, 2007.
 - [69] D. Hulett and D. Hillson. Branching out: Decision trees offer a realistic approach to risk analysis. In *PM Network*. Project Management Institute, 2006.
 - [70] David T. Hulett and David Hillson. Use decision trees to make important project decisions. *Cost Engineering (published by AACEI)*, 2014.
 - [71] Marian V. Iordache and Panos J. Antsaklis. An introduction to petri nets. *Supervisory Control of Concurrent Systems: A Petri Net Structural Approach*, pages 7–28, 2006.
 - [72] Volkan Isler, Sanjeev Khanna, John Spletzer, and Camillo J. Taylor. Target tracking with distributed sensors: The focus of attention problem. *Computer Vision and Image Understanding*, 100(1):225–247, 2005.
 - [73] Teerawat Issariyakul and Ekram Hossain. *Introduction to network simulator NS2*. Springer Science & Business Media, 2011.
 - [74] Anil K. Jain and Richard C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.
 - [75] Bard J. Jarrah A. Pickup and delivery network segmentation using contiguous geographic clustering. *Journal of the Operational Research Society*, advance online publication, 2010.

- [76] Yosr Jarraya, Andrei Soeanu, Mourad Debbabi, and Fawzi Hassaïne. Automatic verification and performance analysis of time-constrained SysML activity diagrams. In *Engineering of Computer Based Systems (ECBS)*, pages 515–522. IEEE Computer Society, 2007.
- [77] Márk Jelasity, Alberto Montresor, and Ozalp Babaoglu. Gossip-based aggregation in large dynamic networks. *ACM Transactions on Computer Systems (TOCS)*, 23:219–252, 2005.
- [78] Xin Jin, Soumalya Sarkar, Asok Ray, Shalabh Gupta, and Thyagaraju Damarla. Target detection and classification using seismic and pir sensors. *IEEE Sensors Journal*, 12(6):1709–1718, 2012.
- [79] G. Kaminka, D. Pynadath, and M. Tambe. Monitoring deployed agent teams. In *Proceedings of the 5th ACM Conference on Autonomous Agents*, pages 308–315, 2001.
- [80] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 24(7):881–892, 2002.
- [81] Daniel Karapetyan and Gregory Gutin. A new approach to population sizing for memetic algorithms: a case study for the multidimensional assignment problem. *Evolutionary computation*, 19(3):345–371, 2011.
- [82] Tarık Küçükdeniz, Alp Baray, Kubilay Ecerkale, and Şakir Esnaf. Integrated use of fuzzy c-means and convex programming for capacitated multi-facility location problem. *Expert Systems with Applications*, 39(4):4306–4314, 2012.
- [83] Yusuke Kuroki and Tomomi Matsui. An approximation algorithm for multidimensional assignment problems minimizing the sum of squared errors. *Discrete Applied Mathematics*, 157(9):2124–2135, 2009.
- [84] M. Kwiatkowska, G. Norman, and D. Parker. Analysis of a gossip protocol in PRISM. *ACM SIGMETRICS Performance Evaluation Review*, 36(3):17–22, 2008.
- [85] Morten Larsen. Branch and bound solution of the multidimensional assignment problem formulation of data association. *Optimization Methods Software*, 27(6):1101–1126, 2012.

- [86] Qingwei Li and Alex Savachkin. A fast tabu search algorithm for the reliable p-median problem. In *Advances in Global Optimization*, pages 417–424. Springer, 2015.
- [87] Shi Li. A 1.488 approximation algorithm for the uncapacitated facility location problem. *Information and Computation*, 222:45–58, 2013.
- [88] Yi-Zhou Li, Hao Hu, and Dao-Zheng Huang. Developing an effective fuzzy logic model for managing risks in marine oil transport. *International Journal of Shipping and Transport Logistics*, 5(4):485–499, 2013.
- [89] Federico Liberatore, Maria P. Scaparra, and Mark S. Daskin. Hedging against disruptions with ripple effects in location analysis. *Omega*, 40(1):21–30, 2012.
- [90] Federico Librino, Marco Levorato, and Michele Zorzi. An algorithmic solution for computing circle intersection areas and its applications to wireless communications. *Wireless Communications and Mobile Computing*, 14(18):1672–1690, 2014.
- [91] Cunrui Ma, Baohua Mao, Qi Xu, Guodong Hua, Sijia Zhang, and Tong Zhang. Multi-depot vehicle routing optimization considering energy consumption for hazardous materials transportation. *Sustainability*, 10(10):3519, 2018.
- [92] Samuel Madden, Michael J. Franklin, Joseph M. Hellerstein, and Wei Hong. Tag: a tiny aggregation service for ad-hoc sensor networks. In *IN OSDI*, 2002.
- [93] Abbas Mahmoudabadi. Developing a chaotic-simulation based model for ranking high selected network links in hazardous material transportation. *Modelling and Simulation in Engineering*, pages 1–9, 2014.
- [94] Ila Manuj and John T. Mentzer. Global supply chain risk management. *Journal of Business Logistics*, 29(1):133–155, 2008.
- [95] James G. March and Zur Shapira. Managerial perspectives on risk and risk taking. *Management science*, 33(11):1404–1418, 1987.
- [96] Hrvoje Markovic, Ivana Cavar, and Tonci Caric. Using data mining to forecast uncertain demands in stochastic vehicle routing problem. ISEP, 2005.

- [97] Alfonso Pedraza Martinez, Sameer Hasija, and Luk Van Wassenhove. An operational mechanism design for fleet management coordination in humanitarian operations. INSEAD Social Innovation Centre, 2010.
- [98] Mujahid Mohsin, Muhammad Usama Sardar, Osman Hasan, and Zahid Anwar. Iotriskanalyzer: A probabilistic model checking based framework for formal risk analytics of the internet of things. *IEEE Access*, 5:5494–5505, 2017.
- [99] Jakub Montewka, Sören Ehlers, Floris Goerlandt, Tomasz Hinz, Kristjan Tabri, and Pentti Kujala. A framework for risk assessment for maritime transportation systems – a case study for open sea collisions involving ropax vessels. *Reliability Engineering & System Safety*, 124:142–157, 2014.
- [100] Damon Mosk-Aoyama and Devavrat Shah. Computing separable functions via gossip. In *Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, pages 113–122, 2006.
- [101] Alper Murat, Gilbert Laporte, and Vedat Verter. A global shooting algorithm for the facility location and capacity acquisition problem on a line with dense demand. *Computers & Operations Research*, 71(Supplement C):1–15, 2016.
- [102] Eduardo Freire Nakamura, Antonio Alfredo Ferreira Loureiro, Azzedine Boukerche, and Albert Y. Zomaya. Localized algorithms for information fusion in resource constrained networks. *Information Fusion*, (15):2–4, 2014.
- [103] Tho Manh Nguyen, Josef Schiefer, and A. Min Tjoa. Sense & response service architecture (saresa): an approach towards a real-time business intelligence solution and its use for a fraud detection application. In *Proceedings of the 8th ACM International Workshop on Data Warehousing and OLAP*, pages 77–86, 2005.
- [104] Charles L. Ortiz, Jr. Introspective and elaborative processes in rational agents. *Annals of Mathematics and Artificial Intelligence*, 25:1–34, 1999.

- [105] Stephan Otto and Gabriella Kokai. Decentralized evolutionary optimization approach to the p-median problem. In *Applications of Evolutionary Computing*, volume 4974, pages 659–668. Springer Berlin Heidelberg, 2008.
- [106] F. Aykut Özsoy and Mustafa Ç. Pinar. An exact algorithm for the capacitated vertex p-center problem. *Computers & Operations Research*, 33(5):1420–1436, 2006.
- [107] Zeynep Özyurt and Deniz Aksen. Solving the multi-depot location-routing problem with lagrangian relaxation. In *Extending the Horizons: Advances in Computing, Optimization, and Decision Technologies*, volume 37, pages 125–144. Springer US, 2007.
- [108] T. Palpanas, R. Sidle, R. Cochrane, and H. Pirahesh. Incremental maintenance for non-distributive aggregate functions. In *Proceedings of the Twenty-Eighth International Conference on Very Large Data Bases*, 2002.
- [109] Min Peng, Yi Peng, and Hong Chen. Post-seismic supply chain risk management: A system dynamics disruption analysis approach for inventory and logistics planning. *Computers & Operations Research*, 42:14–24, 2014.
- [110] Peng Peng, Lawrence V. Snyder, Andrew Lim, and Zuli Liu. Reliable logistics networks design with facility disruptions. *Transportation Research Part B: Methodological*, 45(8):1190–1211, 2011.
- [111] David W. Pentico. Assignment problems: A golden anniversary survey. *European Journal of Operational Research*, 176(2):774–793, 2007.
- [112] William P. Pierskalla. Letter to the editor-the multidimensional assignment problem. *Operations Research*, 16(2):422–431, 1968.
- [113] Aubrey B. Poore and Sabino Gadaleta. Some assignment problems arising from multiple target tracking. *Mathematical and computer modelling*, 43(9):1074–1091, 2006.
- [114] Warren B. Powell. Introduction to Markov Decision Processes. *Approximate Dynamic Programming: Solving the Curses of Dimensionality, Second Edition*, pages 57–109, 2011.

- [115] Liliana Enciso Quispe and Luis Mengual Galan. Behavior of ad hoc routing protocols, analyzed for emergency and rescue scenarios, on a real urban area. *Expert Systems with Applications*, 41(5):2565–2573, 2014.
- [116] Gordana Radivojević and Vladimir Gajović. Supply chain risk modeling by AHP and fuzzy AHP methods. *Journal of Risk Research*, 17(3):337–352, 2014.
- [117] Nina Radojicic, Aleksandar Djenic, and Miroslav Maric. Fuzzy grasp with path relinking for the risk-constrained cash-in-transit vehicle routing problem. *Applied Soft Computing*, 72:486–497, 2018.
- [118] K. Van Raemdonck, C. Macharis, and O. Mairesse. Risk analysis system for the transport of hazardous materials. *Journal of Safety Research*, 45:55–63, 2013.
- [119] Sujoy Ray, Mourad Debbabi, Mohamad Allouche, Nicolas Léchevin, and Micheline Bélanger. Energy-efficient monitor deployment in collaborative distributed setting. *IEEE Transactions on Industrial Informatics*, 12(1):112–123, 2016.
- [120] Sujoy Ray, Andrei Soeanu, Jean Berger, and Mourad Debbabi. The multi-depot split-delivery vehicle routing problem: model and solution algorithm. *Knowledge-Based Systems*, 71:238–265, 2014.
- [121] Sujoy Ray, Andrei Soeanu, Mourad Debbabi, Abdeslem Boukhtouta, and Jean Berger. Modeling multi-depot split-delivery vehicle routing problem. In *In the Proceedings of the IEEE - Conference on Computational Engineering in Systems Applications*, 2012.
- [122] Robbert Van Renesse, Kenneth P. Birman, and Werner Vogels. Astrolabe: A robust and scalable technology for distributed system monitoring, management, and data mining. *ACM Transactions on Computer Systems*, 21:164–206, 2003.
- [123] Anthony J. Richardson, Elizabeth S. Ampt, and Arnim H. Meyburg. *Survey methods for transport planning*. Eucalyptus Press Melbourne, 1995.
- [124] Alex Rogers, Alessandro Farinelli, Ruben Stranders, and Nicholas R. Jennings. Bounded approximate decentralised coordination via the max-sum algorithm. *Artificial Intelligence*, 175(2):730–759, 2011.

- [125] Torbjørn Rundmo, Trond Nordfjærn, Hilde Hestad Iversen, Sigve Oltedal, and Stig H. Jørgensen. The role of risk perception and other risk-related judgements in transportation mode use. *Safety science*, 49(2):226–235, 2011.
- [126] Mohammed Saleh, Andrei Soeanu, Sujoy Ray, Mourad Debbabi, Jean Berger, and Abdeslem Boukhtouta. Mechanism design for decentralized vehicle routing problem. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, pages 749–754. ACM, 2012.
- [127] Vladimir Savic and Santiago Zazo. Cooperative localization in mobile networks using nonparametric variants of belief propagation. *Ad hoc networks*, 11(1):138–150, 2013.
- [128] Zuo-Jun Max Shen, Roger Lezhou Zhan, and Jiawei Zhang. The reliable facility location problem: Formulations, heuristics, and approximation algorithms. *INFORMS Journal on Computing*, 23(3):470–482, 2011.
- [129] Meng Shicong, Kashyap Srinivas Raghav, Venkatramani Chitra, and Liu Ling. Resource-aware application state monitoring. *Parallel and Distributed Systems, IEEE Transactions on*, 23(12):2315–2329, 2012.
- [130] Bitu Sobhani, Matteo Mazzotti, Enrico Paolini, Andrea Giorgetti, and Marco Chiani. Multiple target detection and localization in UWB multistatic radars. In *Ultra-WideBand (ICUWB), 2014 IEEE International Conference on*, pages 135–140. IEEE, 2014.
- [131] Andrei Soeanu, Sujoy Ray, Mourad Debbabi, Jean Berger, and Abdeslem Boukhtouta. A learning based evolutionary algorithm for distributed multi-depot VRP. In *Proceedings of the 16th International Conference on Knowledge-Based Intelligent Information & Engineering Systems (KES)*, 2012.
- [132] Andrei Soeanu, Sujoy Ray, Mourad Debbabi, Jean Berger, Abdeslem Boukhtouta, and Ahmed Ghanmi. A decentralized heuristic for multi-depot split-delivery vehicle routing problem. In *Proceedings of the IEEE ICAL International Conference on Automation and Logistics*. IEEE, 2011.

- [133] Andrei Soeanu, Sujoy Ray, Mourad Debbabi, Jean Berger, Abdeslem Boukhtouta, and Ahmed Ghanmi. Model checking based service delivery planning. In *CCECE*, pages 1–4. IEEE, 2012.
- [134] Leen-Kiat Soh and Costas Tsatsoulis. Reflective negotiating agents for real-time multisensor target tracking. In *International Joint Conference On Artificial Intelligence*, volume 17, pages 1121–1127. Lawrence Erlbaum Associates LTD, 2001.
- [135] Alireza Souri and Nima Jafari Navimipour. Behavioral modeling and formal verification of a resource discovery approach in grid computing. *Expert Systems with Applications*, 41(8):3831–3849, 2014.
- [136] Éfren L. Souza, Eduardo F. Nakamura, and Richard W. Pazzi. Target tracking for sensor networks: A survey. *ACM Computing Surveys (CSUR)*, 49(2):30, 2016.
- [137] Efren Lopes Souza, Andre Campos, and Eduardo Freire Nakamura. Tracking targets in quantized areas with wireless sensor networks. In *Local Computer Networks (LCN), 2011 IEEE 36th Conference on*, pages 235–238. IEEE, 2011.
- [138] Patrick P. A. Storms and Frederik C. R. Spieksma. An LP-based algorithm for the data association problem in multitarget tracking. *Computers & Operations Research*, 30(7):1067–1085, 2003.
- [139] Oncan T., Kabadi S. N., and Nair K. VLSN search algorithms for partitioning problems using matching neighborhoods. *Journal of the Operational Research Society*, pages 388–398, 2008.
- [140] Luca Talarico, Kenneth Sörensen, and Johan Springael. Metaheuristics for the risk-constrained cash-in-transit vehicle routing problem. *European Journal of Operational Research*, 244(2):457–470, 2015.
- [141] Luca Talarico, Johan Springael, Kenneth Sorensen, and Fabio Talarico. A large neighbourhood metaheuristic for the risk-constrained cash-in-transit vehicle routing problem. *Computers & Operations Research*, 78:547–556, 2017.

- [142] Paolo Toth and Daniele Vigo. *The Vehicle Routing Problem*. Society for Industrial and Applied Mathematics (SIAM), 2002.
- [143] Ming-Lang Tseng, Ru-Jen Lin, Yuan-Hsu Lin, Rong-Hui Chen, and Kimhua Tan. Close-loop or open hierarchical structures in green supply chain management under uncertainty. *Expert Systems with Applications*, 41(7):3250–3260, 2014.
- [144] R. Van der Krogt and M. de Weerd. Plan repair as an extension of planning. In *International Conference on Automated Planning and Scheduling*, pages 161–170, 2005.
- [145] Chrysafis Vogiatzis, Eduardo L. Pasiliao, and Panos M. Pardalos. Graph partitions for the multidimensional assignment problem. *Computational Optimization and Applications*, 58(1):205–224, 2014.
- [146] Sonia Waharte and Niki Trigoni. Supporting search and rescue operations with UAVs. In *Emerging Security Technologies (EST), 2010 International Conference on*, pages 142–147. IEEE, 2010.
- [147] Jose L. Walteros, Chrysafis Vogiatzis, Eduardo L. Pasiliao, and Panos M. Pardalos. Integer programming models for the multidimensional assignment problem with star costs. *European Journal of Operational Research*, 235(3):553–568, 2014.
- [148] David E. Wilkins, Thomas J. Lee, and Pauline Berry. Interactive execution monitoring of agent teams. *Journal of Artificial Intelligence Research*, 18:217–261, 2003.
- [149] Fetahi Wuhib, Rolf Stadler, and Mike Spreitzer. Gossip-based resource management for cloud environments. In *Conference on Network and Service Management (CNSM)*, pages 1–8, 2010.
- [150] Xiaolong Xue, Jinfeng Lu, Yaowu Wang, and Qiping Shen. Towards an agent-based negotiation platform for cooperative decision-making in construction supply chain. In *KES-AMSTA*, pages 416–425, 2007.
- [151] Po Yang, Wenyan Wu, Moniri Mansour, and Chibelushi Claude C. Efficient object localization using sparsely distributed passive RFID tags. *IEEE Transactions on Industrial Electronics*, 60(12):5914–5924, 2013.

- [152] B. Yu, Z. Z. Yang, and J. X. Xie. A parallel improved ant colony optimization for multi-depot vehicle routing problem. *Journal of the Operational Research Society*, 62(1):183–188, 2011.
- [153] Jiahong Zhao and Fumin Zhu. A multi-depot vehicle-routing model for the explosive waste recycling. *International Journal of Production Research*, 54(2):550–563, 2016.
- [154] Yi Zhi Zhao and Oon Peen Gan. Distributed design of RFID network for large-scale RFID deployment. In *2006 IEEE International Conference on Industrial Informatics*, pages 44–49, 2006.
- [155] Sedighe Zibaei, Ashkan Hafezalkotob, and Seyed Sajad Ghashami. Cooperative vehicle routing problem: an opportunity for cost saving. *Journal of Industrial Engineering International*, 12(3):271–286, 2016.

Appendix A

.	Sensor	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
Target	Range	40	38	52	40	65	40	45	60	42	55
T1	Location	152,116	143,87	122,102	152,72	186,125	110,81	131,61	180,57	171,85	85,70
T1	152,57		(x)		(-)			(x)	(-)	(x)	
T2	183,114	(x)				(x)			(-)	(x)	
T3	107,93		(x)	(x)			(-)	(-)			(x)
T4	122,72		(x)	(-)	(-)		(x)	(x)	(-)		(-)
T5	91,68			(x)			(x)	(-)			(x)
T6	165,64		(-)		(x)	(-)		(x)	(-)	(x)	
T7	125,121	(x)		(-)		(x)					
T8	183,74				(x)	(x)			(-)	(x)	
T9	76,123			(-)							(x)
T10	171,45				(x)			(x)	(x)	(-)	
T11	168,104	(-)	(x)	(-)	(x)	(x)			(-)	(-)	

Table A.1: Data and target assignment solution for problem P-n23

.	S11	S12
Range	35	45
Location	107,125	88,112
T1		
T2		
T3	(-)	(-)
T4		
T5		(-)
T6		
T7	(x)	(-)
T8		
T9	(x)	(x)
T10		
T11		

Table A.2: Data and target assignment solution for problem P-n23 continued

.	Sensor	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
	Range	50	32	40	40	52	40	40	50	32	40
Target	Location	104,25	110,81	76,123	122,102	174,23	183,114	18,93	134,13	171,45	85,85
T1	88,55	(-)									(x)
T2	85,70	(-)	(x)								(x)
T3	149,34	(x)				(x)			(-)	(x)	
T4	143,87				(x)						
T5	183,74					(-)	(x)			(x)	
T6	168,104						(-)				
T7	91,36	(x)							(-)		
T8	46,114			(x)				(x)			
T9	180,57					(x)				(x)	
T10	116,51	(-)	(x)						(x)		
T11	34,19										
T12	30,74							(x)			
T13	122,72		(x)		(x)						(-)
T14	152,116				(x)		(x)				
T15	107,125			(x)	(-)						
T16	43,102			(-)				(x)			
T17	107,93		(-)		(x)						(x)
T18	15,66							(-)			
T19	152,57					(x)			(-)	(-)	
T20	131,61	(x)	(x)						(-)		

Table A.3: Data and target assignment solution for problem P-n40

.	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20
Range	35	53	50	50	48	45	50	40	40	35
Location	171,85	186,125	88,112	58,83	76,38	9,42	125,121	46,57	55,44	152,72
T1				(-)	(x)				(x)	
T2			(-)	(-)	(-)				(x)	
T3										
T4	(x)						(-)			(x)
T5	(-)	(x)								(-)
T6	(x)	(x)					(x)			
T7					(x)				(x)	
T8			(x)	(-)						
T9	(x)									(-)
T10					(x)					
T11					(x)	(x)		(-)	(x)	
T12				(x)		(-)		(x)	(-)	
T13							(-)			(x)
T14		(-)					(x)			
T15			(x)				(x)			
T16			(x)	(x)						
T17			(x)				(-)			
T18				(x)		(x)		(x)		
T19	(x)									(x)
T20										(x)

Table A.4: Data and target assignment solution for problem P-n40 continued

.	Sensor	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
	Range	42	62	53	45	62	55	51	51	53	43
Target	Location	116,51	9,6	134,13	104,25	107,125	34,19	9,42	125,121	183,114	76,38
T1	85,23	(-)		(-)	(x)		(x)				(-)
T2	122,72	(x)				(-)			(-)		
T3	122,102					(-)			(x)		
T4	171,85										
T5	30,74							(x)			
T6	183,74									(-)	
T7	88,112					(x)			(-)		
T8	171,45			(x)							
T9	46,57						(x)	(-)			(x)
T10	15,66		(x)				(x)	(-)			
T11	55,44		(-)				(-)	(x)			(x)
T12	131,61	(-)		(x)	(x)						
T13	152,116					(x)			(x)	(x)	
T14	143,87					(-)			(x)	(x)	
T15	85,70	(x)				(-)					(x)
T16	91,36	(x)		(x)	(-)						(-)
T17	24,27		(x)				(x)	(x)			
T18	85,85					(x)					
T19	149,34	(-)		(x)							
T20	110,81	(x)				(-)			(x)		
T21	76,123					(x)			(-)		
T22	18,93										

Table A.5: Data and target assignment solution for problem P-n45

.	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20
Range	33	62	41	48	37	37	49	43	42	54
Location	152,57	9,116	58,13	46,114	152,72	180,57	168,104	58,83	107,93	174,23
T1			(x)							
T2					(x)				(x)	
T3							(x)		(x)	
T4					(x)	(x)	(-)			
T5		(-)		(-)				(x)		
T6					(-)	(-)	(x)			(x)
T7				(x)				(-)	(x)	
T8	(x)				(-)	(x)				(-)
T9								(-)		
T10		(x)								
T11			(x)					(-)		
T12	(-)				(x)				(-)	
T13							(-)			
T14	(x)				(-)		(-)		(-)	
T15								(x)	(-)	
T16			(x)							
T17			(-)							
T18								(-)	(-)	
T19	(x)									(x)
T20									(x)	
T21				(x)						
T22		(x)		(x)				(x)		

Table A.6: Data and target assignment solution for problem P-n45 continued-1

.	S21	S22	S23
Range	52	51	52
Location	88,55	43,102	186,125
T1	(-)		
T2	(-)		
T3			
T4			(x)
T5		(x)	
T6			(x)
T7		(-)	
T8			
T9	(-)	(x)	
T10		(-)	
T11	(-)		
T12	(-)		
T13			(-)
T14			
T15	(-)		
T16	(-)		
T17			
T18	(x)	(x)	
T19			
T20	(-)		
T21		(x)	
T22		(-)	

Table A.7: Data and target assignment solution for problem P-n45 continued-2

.	Sensor	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
Range		49	33	41	50	38	43	37	36	38	36
Target	Location	128,19	146,70	61,36	146,23	27,47	58,85	125,44	95,59	146,51	43,30
T1	30,66					(x)	(-)				
T2	140,34	(x)			(x)			(-)		(x)	
T3	40,21			(x)		(-)					(x)
T4	119,81		(-)						(x)		
T5	101,25	(x)			(x)			(-)	(-)		
T6	159,13	(x)			(x)						
T7	61,95						(-)				
T8	149,74		(x)							(-)	
T9	85,108					(x)					
T10	137,119										
T11	116,70		(x)					(x)	(-)	(x)	
T12	162,89		(-)								
T13	131,61	(-)	(-)		(-)			(x)		(-)	
T14	58,80						(x)				
T15	58,63			(x)		(-)	(-)				
T16	104,44	(-)			(x)			(x)	(-)		
T17	177,23				(x)						
T18	183,61										
T19	85,89						(-)		(x)		
T20	73,49			(x)			(-)		(x)		(-)
T21	183,40				(-)						
T22	88,138										
T23	146,89		(x)							(-)	
T24	95,78						(x)		(x)		
T25	30,27			(-)		(x)					(x)

Table A.8: Data and target assignment solution for problem P-n50

.	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20
Range	53	45	43	58	33	36	50	57	58	42
Location	73,19	162,80	195,21	46,116	162,59	73,106	101,91	183,85	183,102	152,44
T1				(-)						
T2										(-)
T3	(x)									
T4		(x)					(-)			
T5	(x)									
T6			(-)							(-)
T7				(-)		(x)	(-)			
T8		(x)			(-)			(-)	(-)	(x)
T9				(x)		(-)	(-)			
T10							(-)		(x)	
T11							(-)			
T12		(x)			(x)			(-)	(x)	
T13		(x)			(-)		(-)			(x)
T14				(-)		(-)	(-)			
T15	(-)			(x)						
T16	(x)						(-)			
T17			(x)							(x)
T18		(-)	(-)		(x)			(-)	(x)	(x)
T19				(x)		(x)	(-)			
T20	(x)									
T21			(-)		(-)			(x)		(x)
T22				(x)		(x)	(-)			
T23		(x)					(-)	(-)	(-)	
T24						(-)	(x)			
T25	(x)									

Table A.9: Data and target assignment solution for problem P-n50 continued-1

.	S21	S22	S23	S24	S25
Range	58	43	39	58	58
Location	21,100	162,32	116,119	15,76	162,117
T1	(x)			(x)	
T2		(-)			
T3					
T4			(-)		(x)
T5					
T6		(x)			
T7	(x)			(x)	
T8					(-)
T9			(x)		
T10			(x)		(x)
T11					
T12					(-)
T13		(-)			
T14	(x)			(x)	
T15	(-)			(x)	
T16					
T17		(-)			
T18		(-)			
T19					
T20					
T21		(x)			
T22			(x)		
T23					(x)
T24					
T25				(-)	

Table A.10: Data and target assignment solution for problem P-n50 continued-2

.	Sensor	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
	Range	45	43	38	45	46	38	60	56	53	33
Target	Location	46,57	143,87	180,57	140,47	91,36	88,112	9,6	125,121	122,72	85,70
T1	107,125						(x)		(x)		
T2	152,72		(-)	(-)	(-)				(-)	(x)	
T3	91,68					(-)				(-)	(-)
T4	131,61		(x)		(-)					(-)	
T5	85,85						(-)		(-)	(-)	(x)
T6	18,93										
T7	76,123						(x)		(x)		
T8	43,102										
T9	15,66	(x)									
T10	152,57		(x)	(-)	(x)					(-)	
T11	149,34				(x)					(-)	
T12	183,74		(x)	(x)							
T13	58,13					(x)		(-)			
T14	76,38	(-)				(x)					
T15	30,74	(-)									
T16	122,102		(x)				(-)		(x)	(-)	
T17	174,23			(-)	(x)						
T18	24,27	(x)						(x)			
T19	113,13				(-)	(x)					
T20	55,44	(x)				(x)		(-)			
T21	171,85		(-)	(-)						(-)	
T22	110,81		(-)						(-)	(-)	(-)
T23	168,104		(x)						(-)		
T24	104,25				(x)	(-)				(-)	
T25	70,55	(-)				(x)					(-)

Table A.11: Data and target assignment solution for problem P-n51

.	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20
Range	52	56	31	65	39	35	48	48	53	40
Location	183,114	9,42	88,55	186,125	171,45	116,51	152,116	134,13	46,114	85,23
T1							(x)			
T2				(x)	(-)		(-)			
T3			(x)			(-)				
T4						(x)				
T5			(-)						(-)	
T6		(-)							(x)	
T7									(x)	
T8									(x)	
T9		(-)								
T10					(-)			(-)		
T11					(x)			(-)		
T12	(-)			(x)	(-)					
T13										(x)
T14			(x)							(x)
T15		(x)							(x)	
T16							(-)			
T17					(x)			(x)		
T18		(-)								
T19								(x)		(x)
T20		(-)								(x)
T21	(-)			(x)			(x)			
T22						(x)				
T23	(x)			(-)			(x)			
T24						(x)		(-)		(x)
T25			(-)							(-)

Table A.12: Data and target assignment solution for problem P-n51 continued-1

.	S21	S22	S23	S24	S25	S26
Range	60	46	39	36	45	60
Location	9,116	58,83	165,64	107,93	70,98	34,19
T1				(-)		
T2			(x)			
T3		(-)		(x)	(x)	
T4			(x)			
T5		(-)		(x)	(x)	
T6	(x)	(x)				
T7		(-)			(-)	
T8	(-)	(x)			(x)	
T9	(x)					(x)
T10			(x)			
T11			(x)			
T12			(-)			
T13						(x)
T14						(-)
T15	(-)	(x)				(-)
T16				(x)		
T17						
T18						(x)
T19						
T20		(-)				(-)
T21			(x)			
T22				(x)	(x)	
T23						
T24						
T25		(x)			(x)	(-)

Table A.13: Data and target assignment solution for problem P-n51 continued-2

.	Sensor	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
Range	34	47	45	42	50	45	53	39	56	33	
Location	183,61	58,63	61,36	146,89	159,13	40,100	21,100	104,44	195,21	183,85	
T1	198,72	(-)								(x)	(x)
T2	61,95		(x)				(x)	(-)			
T3	128,19					(x)			(x)		
T4	162,89				(-)						(-)
T5	162,59	(-)			(x)	(-)				(-)	
T6	162,32					(x)				(-)	
T7	88,138										
T8	131,61				(-)				(-)		
T9	58,85		(-)				(-)	(x)			
T10	73,106		(-)				(-)	(-)			
T11	162,117				(x)						
T12	183,40	(x)				(x)				(x)	
T13	82,68		(-)	(-)					(x)		
T14	146,51				(-)	(-)					
T15	146,23					(x)				(x)	
T16	101,91										
T17	43,30		(-)	(x)							
T18	119,81				(x)						
T19	159,66	(-)			(x)						(-)
T20	73,19		(x)	(-)							
T21	73,49		(x)	(x)					(-)		
T22	162,102				(x)						(-)
T23	27,47		(-)	(-)							
T24	149,74				(-)						
T25	15,76		(x)				(-)	(x)			
T26	140,34					(x)			(x)		
T27	85,89		(x)								

Table A.14: Data and target assignment solution for problem P-n55

.	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20
Range	26	46	33	35	53	36	49	47	50	40
Location	125,44	177,23	146,70	40,21	137,119	95,59	46,116	183,102	116,119	85,108
T1								(x)		
T2							(-)			(x)
T3	(-)									
T4			(x)		(x)			(x)		
T5		(-)	(-)							
T6		(x)								
T7					(x)		(x)		(x)	(-)
T8	(x)		(x)							
T9							(-)			(x)
T10							(x)		(-)	(x)
T11					(x)			(x)	(-)	
T12		(-)								
T13						(x)				
T14	(x)	(-)	(x)							
T15		(-)								
T16					(x)	(x)			(-)	(x)
T17				(-)						
T18			(-)		(-)	(-)			(x)	
T19			(-)					(x)		
T20				(x)						
T21						(x)				
T22					(x)			(x)	(-)	
T23				(x)						
T24			(x)		(-)			(-)		
T25										
T26	(x)	(-)								
T27						(x)	(-)		(-)	(x)

Table A.15: Data and target assignment solution for problem P-n55 continued-1

.	S21	S22	S23	S24	S25	S26	S27	S28
Range	41	36	46	34	36	26	36	51
Location	101,25	162,80	30,66	58,80	95,78	152,44	116,70	30,27
T1								
T2			(-)	(-)				
T3	(x)							
T4		(-)						
T5		(x)				(x)		
T6						(x)		
T7								
T8							(x)	
T9			(-)	(x)				
T10				(-)	(x)			
T11								
T12								
T13				(x)	(-)		(-)	
T14		(x)				(-)	(-)	
T15						(x)		
T16					(-)		(-)	
T17			(x)					(x)
T18					(-)		(x)	
T19		(-)				(x)		
T20	(x)							(-)
T21	(-)							(-)
T22		(-)						
T23			(x)					(x)
T24		(x)					(x)	
T25			(x)					
T26	(-)					(-)		
T27				(-)	(-)			

Table A.16: Data and target assignment solution for problem P-n55 continued-2

.	Sensor	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
	Range	38	48	38	54	49	43	31	37	39	52
Target	Location	30,27	85,108	177,23	24,127	73,49	88,138	131,61	95,59	119,81	159,13
T1	61,95		(-)		(x)	(x)					
T2	183,85										
T3	162,89										
T4	58,63					(x)					
T5	30,66					(x)					
T6	198,72										
T7	15,76				(x)						
T8	46,116		(x)		(x)						
T9	116,108		(-)				(-)			(x)	
T10	146,2			(x)							(x)
T11	104,44					(-)			(x)		
T12	149,74							(x)		(-)	
T13	116,119		(-)				(x)			(x)	
T14	73,106		(-)		(-)		(x)				
T15	162,32			(x)							(-)
T16	159,66							(x)			
T17	82,68		(-)			(-)			(x)		
T18	183,40			(-)							(x)
T19	73,19					(-)					
T20	95,78		(-)			(x)			(x)	(-)	
T21	146,89									(-)	
T22	128,19										(x)
T23	43,30	(-)				(-)					
T24	40,100		(-)		(x)						
T25	162,59										(-)
T26	58,85		(-)			(x)					
T27	162,102										
T28	12,42	(x)									
T29	152,44			(x)				(-)			(-)
T30	116,70					(-)		(x)	(-)	(x)	

Table A.17: Data and target assignment solution for problem P-n60

.	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20
Range	46	53	59	52	50	37	24	38	19	45
Location	137,119	61,36	21,100	195,21	40,21	140,34	146,51	125,44	146,70	162,80
T1			(-)							
T2										(x)
T3	(-)									(x)
T4		(-)	(-)		(x)					
T5		(-)	(x)		(-)					
T6				(x)						(x)
T7			(x)							
T8			(-)							
T9	(x)									
T10						(-)				
T11		(-)						(x)		
T12							(x)		(x)	(-)
T13	(x)									
T14			(x)							
T15				(-)		(x)				
T16							(x)		(-)	(-)
T17		(x)								
T18				(x)						
T19		(x)			(x)					
T20										
T21	(x)								(x)	(x)
T22						(-)		(x)		
T23		(x)			(x)					
T24			(x)							
T25				(x)		(x)	(-)			(-)
T26		(x)	(-)							
T27	(-)									(x)
T28		(x)	(x)		(-)					
T29				(x)		(-)	(x)	(-)		(-)
T30								(x)		

Table A.18: Data and target assignment solution for problem P-n60 continued-1

.	S21	S22	S23	S24	S25	S26	S27	S28	S29	S30
Range	40	44	33	51	44	37	44	28	43	37
Location	192,45	101,91	183,102	101,25	58,80	27,47	146,23	183,61	162,117	85,89
T1		(x)			(-)					(-)
T2			(-)					(x)	(x)	
T3			(x)						(x)	
T4					(x)	(-)				
T5					(-)	(x)				
T6	(x)							(-)		
T7					(-)	(x)				
T8					(x)					
T9		(-)								(x)
T10				(x)			(-)			
T11				(x)						
T12										
T13		(-)								
T14		(-)			(-)					(x)
T15	(x)						(-)			
T16	(-)							(x)		
T17		(-)		(-)	(x)					(-)
T18	(x)						(-)	(-)		
T19				(x)						
T20		(x)			(-)					(-)
T21									(-)	
T22				(-)			(x)			
T23						(x)				
T24					(x)					
T25	(-)						(-)	(x)		
T26		(-)			(x)					(-)
T27			(x)						(x)	
T28						(-)				
T29							(-)			
T30		(-)		(-)						(-)

Table A.19: Data and target assignment solution for problem P-n60 continued-2

.	Sensor	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
Range		39	51	38	52	77	46	34	37	46	56
Target	Location	85,108	159,13	61,36	30,66	88,138	40,21	101,25	101,91	146,70	162,117
T1	15,76				(x)						
T2	46,116					(-)					
T3	162,80									(-)	(-)
T4	162,102									(-)	(x)
T5	40,4						(x)				
T6	104,6							(x)			
T7	192,45		(x)								
T8	119,81					(x)			(x)	(x)	
T9	27,47			(-)	(-)		(-)				
T10	82,68					(-)			(x)		
T11	85,32			(x)				(-)			
T12	85,89	(-)				(-)			(x)		
T13	146,51		(-)							(x)	
T14	116,108	(x)				(x)			(-)		(x)
T15	177,23		(x)								
T16	162,89									(-)	(x)
T17	183,102										(x)
T18	146,23		(x)								
T19	95,59								(x)		
T20	58,80	(-)			(x)	(-)					
T21	125,44		(-)					(x)		(-)	
T22	162,59		(x)							(-)	
T23	149,74									(x)	(-)
T24	195,21		(x)								
T25	95,78	(-)				(x)			(-)		
T26	30,27			(x)	(-)		(-)				
T27	58,63			(-)	(-)		(-)				
T28	152,44		(x)							(-)	
T29	137,119					(x)					(x)
T30	183,61									(-)	
T31	73,106	(-)				(x)			(-)		
T32	131,61									(-)	

Table A.20: Data and target assignment solution for problem P-n65

.	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20
Range	42	33	61	30	60	23	33	42	53	41
Location	43,30	183,40	116,119	140,34	207,116	159,66	183,85	104,44	73,19	73,49
T1										
T2										
T3			(x)		(x)	(x)	(-)			
T4			(x)		(-)		(-)			
T5	(x)								(-)	
T6								(-)	(x)	
T7		(-)								
T8			(-)					(-)		
T9	(-)									
T10								(-)	(-)	(-)
T11								(-)	(x)	(x)
T12			(-)							
T13				(x)		(-)				
T14			(-)							
T15		(-)								
T16			(-)		(-)		(-)			
T17					(x)		(-)			
T18				(-)						
T19								(-)	(x)	(x)
T20										(-)
T21				(-)				(x)		
T22		(-)				(x)				
T23			(x)			(x)				
T24		(x)								
T25			(-)					(x)		(-)
T26	(x)								(-)	
T27	(-)								(-)	(-)
T28		(x)		(x)						
T29			(x)							
T30		(x)					(x)			
T31			(-)							
T32			(-)	(x)				(x)		

Table A.21: Data and target assignment solution for problem P-n65 continued-1

.	S21	S22	S23	S24	S25	S26	S27	S28	S29	S30
Range	51	55	35	33	56	41	33	49	40	34
Location	189,2	12,42	55,51	58,85	40,100	162,32	116,70	198,72	146,89	128,19
T1		(-)			(-)					
T2					(x)					
T3								(-)	(-)	
T4								(x)	(-)	
T5		(x)								
T6										(x)
T7	(x)					(-)		(x)		
T8							(-)		(-)	
T9		(x)	(x)		(x)					
T10			(x)	(-)	(-)					
T11										
T12				(x)	(-)					
T13						(x)			(-)	
T14									(-)	
T15	(x)					(x)				
T16								(x)	(x)	
T17								(x)	(-)	
T18	(-)					(x)				(x)
T19							(-)			
T20			(-)	(x)	(-)					
T21						(x)	(-)			(-)
T22						(x)		(-)	(-)	
T23									(-)	
T24	(x)					(-)				
T25							(-)			
T26		(x)	(-)							
T27		(-)	(x)	(x)	(-)					
T28						(-)				
T29									(-)	
T30						(x)		(-)		
T31				(-)	(x)					
T32							(x)		(-)	

Table A.22: Data and target assignment solution for problem P-n65 continued-2

.	S31	S32	S33
Range	44	53	62
Location	61,95	21,100	24,127
T1		(x)	(x)
T2	(x)	(-)	(x)
T3			
T4			
T5			
T6			
T7			
T8			
T9			
T10	(x)		
T11			
T12	(x)		
T13			
T14			
T15			
T16			
T17			
T18			
T19			
T20	(-)	(x)	(-)
T21			
T22			
T23			
T24			
T25	(x)		
T26			
T27	(-)	(x)	
T28			
T29			
T30			
T31	(x)	(-)	(-)
T32			

Table A.23: Data and target assignment solution for problem P-n65 continued-3

.	Sensor	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
Range		47	48	56	26	55	39	47	27	22	41
Target	Location	116,119	46,116	24,127	162,59	40,4	73,19	192,45	131,74	131,61	189,2
T1		61,95	(x)	(x)							
T2		162,89									
T3		146,127	(x)								
T4		183,61			(x)			(x)			
T5		30,66									
T6		116,70							(-)	(-)	
T7		152,44			(-)			(x)			
T8		162,32						(x)			(x)
T9		12,42				(x)					
T10		183,40						(-)			(-)
T11		146,2									
T12		125,44								(-)	
T13		119,81	(-)						(x)		
T14		61,36				(-)	(x)				
T15		146,70			(-)				(-)	(x)	
T16		85,108	(-)	(x)							
T17		27,47				(x)					
T18		162,117	(x)								
T19		82,68									
T20		159,13						(-)			(x)
T21		159,66			(x)			(-)			
T22		73,49					(-)				
T23		137,119	(x)								
T24		58,85		(x)	(-)						
T25		146,23									
T26		85,32				(-)	(x)				
T27		110,57								(x)	
T28		58,63									
T29		116,108	(x)								
T30		149,74			(x)				(-)		
T31		21,100		(-)	(x)						
T32		198,72						(x)			
T33		162,102									
T34		88,138	(x)	(x)							
T35		104,44									

Table A.24: Data and target assignment solution for problem P-n70

.	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20
Range	24	40	26	60	29	32	47	34	35	50
Location	146,51	73,106	58,80	40,21	95,59	85,89	104,6	128,19	40,100	101,91
T1		(-)	(x)			(-)			(-)	(-)
T2										
T3										
T4										
T5				(-)						
T6					(-)					(-)
T7	(x)									
T8										
T9				(-)						
T10										
T11							(x)	(-)		
T12	(x)						(-)	(-)		
T13										(x)
T14				(-)						
T15	(-)									(-)
T16		(x)				(-)				(x)
T17				(-)						
T18										
T19		(-)			(x)	(x)				(-)
T20								(x)		
T21	(-)									
T22				(-)	(x)					
T23										(-)
T24		(-)	(-)			(-)			(-)	(x)
T25							(-)	(x)		
T26				(-)	(x)		(x)			
T27					(-)					(-)
T28			(x)	(x)						
T29										(x)
T30	(-)									
T31									(x)	
T32										
T33										
T34		(-)								(x)
T35					(x)		(x)			(x)

Table A.25: Data and target assignment solution for problem P-n70 continued-1

.	S21	S22	S23	S24	S25	S26	S27	S28	S29	S30
Range	56	38	38	45	34	52	48	35	48	28
Location	195,21	43,30	146,89	183,85	55,51	207,116	183,102	95,78	101,25	140,34
T1										
T1			(-)	(x)			(x)			
T2			(-)				(x)			
T3	(-)			(x)			(-)			
T4					(x)					
T5			(x)					(x)	(x)	
T6	(x)									(-)
T7	(-)									(x)
T8		(-)								
T9	(x)			(x)						
T10	(x)									
T11									(x)	(x)
T12			(x)					(-)		
T13		(-)			(x)				(x)	
T14			(-)	(x)						
T15								(-)		
T16		(-)			(-)					
T17			(-)	(-)		(x)	(x)			
T18					(-)			(-)	(x)	
T19	(x)									
T20			(-)	(-)			(x)			
T21		(x)			(-)				(x)	
T22			(x)							
T23										
T24	(x)								(-)	(-)
T25									(-)	
T26								(x)	(x)	
T27		(-)			(x)					
T28			(x)							
T29			(x)	(-)			(-)			
T30										
T31	(-)			(x)		(x)	(-)			
T32			(x)	(x)		(-)	(x)			
T33										
T34									(-)	

Table A.26: Data and target assignment solution for problem P-n70 continued-2

.	S31	S32	S33	S34	S35
Range	45	63	58	54	33
Location	177,23	30,27	168,131	15,76	162,80
T1				(-)	
T2			(-)		(x)
T3			(x)		
T4	(-)				(-)
T5		(x)		(x)	
T6					
T7	(-)				
T8	(-)				
T9		(x)		(x)	
T10	(x)				
T11	(x)				
T12					
T13					
T14		(-)			
T15					(x)
T16					
T17		(x)		(x)	
T18			(-)		
T19					
T20	(-)				
T21					(x)
T22		(-)			
T23			(x)		
T24				(x)	
T25	(x)				
T26		(-)			
T27					
T28		(-)		(-)	
T29			(-)		
T30					(x)
T31				(x)	
T32					
T33			(-)		(-)
T34					
T35					

Table A.27: Data and target assignment solution for problem P-n70 continued-3

.	Sensor	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
	Range	31	18	28	50	44	51	60	26	29	41
Target	Location	152,44	131,61	116,108	27,47	82,68	195,9	40,21	177,23	24,127	146,23
T1	15,76				(x)						
T2	146,89										
T3	183,40						(-)		(x)		(x)
T4	85,108					(x)					
T5	21,100									(x)	
T6	116,70		(-)			(x)					
T7	137,119			(-)							
T8	85,32					(-)		(x)			
T9	183,61										
T10	85,89					(x)					
T11	30,27				(-)			(x)			
T12	61,95					(-)					
T13	116,119			(x)							
T14	58,63				(-)	(x)		(x)			
T15	128,19										(x)
T16	46,116									(-)	
T17	95,78					(x)					
T18	195,21						(x)		(x)		
T19	110,57					(-)					
T20	162,89										
T21	183,85										
T22	101,108			(x)							
T23	131,74		(x)								
T24	189,2						(x)		(-)		
T25	183,102										
T26	125,44	(x)									(x)
T27	146,51	(x)									(-)
T28	162,32	(-)					(-)		(-)		(x)
T29	58,80				(x)	(-)					
T30	168,131										
T31	192,45						(-)				
T32	101,91			(-)		(x)					
T33	61,36				(-)	(-)		(x)			
T34	159,66	(x)									
T35	116,32										(-)
T36	162,80										
T37	55,51				(x)	(-)		(-)			
T38	73,19							(-)			

Table A.28: Data and target assignment solution for problem P-n76

.	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20
Range	36	28	57	46	41	33	55	42	23	43
Location	43,30	95,59	30,66	146,2	162,102	104,6	146,70	88,138	76,40	73,106
T1			(-)							
T2					(x)		(-)			
T3							(-)			
T4								(x)		(-)
T5			(x)							
T6		(-)					(-)			
T7					(x)		(-)			
T8						(-)			(x)	
T9							(-)			
T10										(-)
T11	(-)		(-)							
T12			(x)							(x)
T13								(-)		
T14			(x)							
T15				(-)		(x)	(-)			
T16			(x)							(x)
T17		(-)					(-)			(-)
T18										
T19		(x)					(-)			
T20					(x)		(-)			
T21					(x)		(-)			
T22								(x)		(-)
T23							(-)			
T24				(x)						
T25					(x)		(-)			
T26							(-)			
T27							(-)			
T28				(x)			(-)			
T29			(-)							(-)
T30					(-)					
T31							(x)			
T32							(-)			(-)
T33	(-)		(-)						(x)	
T34					(x)		(-)			
T35				(-)		(x)	(-)			
T36					(x)		(-)			
T37	(-)		(-)							
T38	(x)								(-)	

Table A.29: Data and target assignment solution for problem P-n76 continued-1

.	S21	S22	S23	S24	S25	S26	S27	S28	S29	S30
Range	44	51	48	31	43	37	40	61	35	30
Location	58,85	198,72	174,4	162,59	104,44	149,74	159,13	12,42	119,81	101,25
T1	(-)							(x)		
T2						(x)			(x)	
T3		(-)	(-)	(x)			(-)			
T4	(x)									
T5	(-)							(x)		
T6					(-)	(x)			(-)	
T7										
T8					(-)					(x)
T9		(x)		(x)		(x)				
T10	(x)								(x)	
T11								(x)		
T12	(x)									
T13										
T14	(-)							(-)		
T15					(-)		(-)			(-)
T16	(-)									
T17	(x)				(-)				(x)	
T18			(-)				(x)			
T19					(x)				(-)	
T20		(-)		(x)		(-)				
T21		(x)				(-)				
T22									(x)	
T23					(-)	(x)			(-)	
T24			(x)				(-)			
T25		(x)								
T26					(x)					
T27				(-)	(-)	(x)				
T28			(-)	(x)			(-)			
T29	(x)							(x)		
T30										
T31		(x)	(x)							
T32	(x)								(x)	
T33								(x)		
T34		(-)		(x)		(-)				
T35					(-)					(x)
T36		(-)		(x)		(x)				
T37	(-)							(-)		
T38					(-)					(x)

Table A.30: Data and target assignment solution for problem P-n76 continued-2

.	S31	S32	S33	S34	S35	S36	S37	S38
Range	36	29	51	44	47	25	41	41
Location	140,34	73,49	40,4	207,116	162,117	116,64	40,100	146,127
T1							(x)	
T2					(-)			(-)
T3								
T4								
T5							(-)	
T6						(x)		
T7					(x)			(x)
T8		(-)						
T9								
T10								
T11			(x)					
T12							(-)	
T13					(x)			(x)
T14		(-)						
T15	(x)							
T16							(x)	
T17								
T18								
T19						(x)		
T20					(x)			
T21				(-)	(x)			
T22								
T23						(x)		
T24								
T25				(-)	(x)			
T26	(-)					(-)		
T27	(x)							
T28	(-)							
T29							(-)	
T30				(x)	(x)			(x)
T31								
T32								
T33		(-)	(-)					
T34								
T35	(x)							
T36					(-)			
T37		(x)	(x)					
T38			(x)					

Table A.31: Data and target assignment solution for problem P-n76 continued-3

	Sensor	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
	Range	79	147	131	79	136	65	100	68	83	132
Target	Location	261,182	485,13	351,455	321,110	187,429	172,136	396,273	358,208	299,26	209,13
T1	104,123										
T2	97,175										
T3	470,110		(x)								
T4	403,234							(x)	(-)		
T5	284,370			(-)		(-)					
T6	440,318							(-)			
T7	470,338							(x)			
T8	209,370					(x)					
T9	246,429			(x)		(x)					
T10	351,52		(-)		(-)					(-)	
T11	381,318							(x)			
T12	433,58		(x)								
T13	134,110						(-)				(x)
T14	470,208							(-)			
T15	455,403			(-)							
T16	179,318					(-)					
T17	321,45				(-)					(x)	(-)
T18	381,260							(x)	(-)		
T19	172,117						(x)				(x)
T20	291,299										
T21	313,91				(x)					(-)	(x)
T22	209143	(-)					(-)				(x)
T23	22,175										
T24	246,208	(x)									
T25	0,370										
T26	149,156						(-)				
T27	82318										
T28	142,136						(-)				
T29	67,71										
T30	179156						(x)				
T31	410,169								(-)		
T32	60,110										
T33	216,318					(-)					
T34	246,91				(-)						(x)
T35	30,422										
T36	134,240										
T37	90,221										
T38	410,292							(x)			
T39	396,331							(-)			
T40	284,143	(x)			(x)						
T41	97,286										
T42	119,136						(-)				
T43	149,123						(x)				(x)
T44	291,221	(-)									
T45	216,416					(x)					
T46	164,58										(-)
T47	336,84				(x)					(x)	
T48	224,58									(x)	(-)
T49	351,357			(x)				(x)			
T50	321,175	(-)			(-)				(x)		

Table A.32: Data and target assignment solution for problem P-n101

.	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20
Range	70	94	87	85	85	84	114	113	129	142
Location	75,136	119,97	261,286	381,58	112,201	261,344	455,130	15,97	164,357	134,403
T1	(x)	(-)			(-)			(-)		
T2	(-)	(x)			(x)					
T3							(x)			
T4										
T5						(x)		(x)		
T6										
T7										
T8						(x)		(-)	(-)	
T9								(x)	(-)	
T10				(-)						
T11										
T12				(x)		(x)				
T13	(-)	(x)								
T14						(x)				
T15										
T16								(x)	(-)	
T17				(-)						
T18										
T19		(x)								
T20			(x)			(-)				
T21				(x)						
T22										
T23	(x)							(x)		
T24			(-)							
T25										(-)
T26		(-)			(x)					
T27								(-)	(x)	
T28	(-)	(-)			(x)					
T29	(x)	(-)						(-)		
T30		(x)			(-)					
T31							(-)			
T32	(x)	(-)						(x)		
T33			(x)			(-)			(x)	(-)
T34										
T35										(x)
T36					(x)			(x)		
T37					(x)					
T38										
T39										
T40										
T41								(-)	(x)	
T42	(-)	(x)			(-)			(-)		
T43		(-)								
T44			(x)							
T45								(x)	(x)	
T46		(-)								
T47				(x)						
T48										
T49										
T50										

Table A.33: Data and target assignment solution for problem P-n101 continued-1

.	S21	S22	S23	S24	S25	S26	S27	S28	S29	S30
Range	105	165	77	148	123	127	75	99	123	155
Location	194,97	97,481	172,175	22,13	157,0	396,13	254,149	134,305	410,422	448,481
T1	(-)			(x)						
T2			(-)							
T3						(-)				
T4										
T5										
T6									(x)	
T7									(x)	(-)
T8		(-)								
T9		(-)								
T10						(x)				
T11									(x)	
T12						(-)				
T13	(-)		(-)		(-)					
T14										
T15									(x)	(x)
T16								(-)		
T17						(x)				
T18										
T19	(-)		(-)		(-)					
T20										
T21						(-)				
T22	(-)		(x)				(x)			
T23										
T24							(-)			
T25		(x)								
T26	(-)		(-)							
T27		(-)						(x)		
T28	(-)		(x)							
T29				(x)	(-)					
T30	(-)		(-)							
T31										
T32				(-)						
T33								(-)		
T34	(-)						(x)			
T35		(x)								
T36			(-)					(-)		
T37								(-)		
T38										
T39									(-)	
T40	(x)						(-)			
T41								(-)		
T42	(-)		(-)							
T43	(-)		(-)							
T44										
T45		(-)								
T46	(x)				(x)					
T47						(-)				
T48	(x)				(x)					
T49									(-)	
T50							(x)			

Table A.34: Data and target assignment solution for problem P-n101 continued-2

.	S31	S32	S33	S34	S35	S36	S37	S38	S39	S40
Range	110	72	109	93	81	92	130	113	97	80
Location	351,253	67,182	328,65	97,45	396,370	60,260	45,344	463,253	0,292	179,208
T1		(-)		(-)						
T2		(x)								
T3										
T4	(x)							(-)		
T5										
T6					(-)			(x)		
T7					(-)			(x)		
T8										
T9										
T10			(x)							
T11	(-)				(-)			(-)		
T12			(-)							
T13				(-)						
T14								(x)		
T15					(x)					
T16										
T17			(x)							
T18	(x)							(-)		
T19										
T20	(x)									
T21			(-)							
T22										(-)
T23		(-)								
T24										(-)
T25							(-)		(x)	
T26										(x)
T27						(-)	(x)		(-)	
T28										
T29				(x)						
T30										(x)
T31	(x)							(-)		
T32				(x)						
T33										
T34			(x)							
T35							(x)			
T36						(-)				(-)
T37		(x)				(x)				
T38	(-)				(-)			(x)		
T39	(x)				(x)			(x)		
T40			(-)							
T41						(-)	(x)			
T42		(-)								
T43										
T44	(x)									
T45										
T46				(x)						
T47			(-)							
T48			(-)							
T49	(-)				(-)					

Table A.35: Data and target assignment solution for problem P-n101 continued-3

.	S41	S42	S43	S44	S45	S46	S47	S48	S49	S50	S51
Range	69	75	124	84	101	85	84	107	103	104	138
Location	97,104	134,149	97,370	336,286	30,227	396,110	127,117	403,221	246,240	187,260	321,403
T1	(-)	(-)					(x)				
T2		(-)			(-)		(-)				
T3						(x)					
T4								(x)			
T5											(x)
T6								(x)			
T7											
T8			(x)								(-)
T9											(-)
T10						(x)					
T11				(x)				(-)			(-)
T12						(-)					
T13	(-)	(-)					(x)				
T14								(x)			
T15											(-)
T16			(x)						(x)	(-)	
T17											
T18				(-)				(x)			
T19		(-)					(-)				
T20				(x)					(-)		(-)
T21											
T22											
T23					(x)						
T24									(x)	(x)	
T25			(x)								
T26		(-)					(x)				
T27			(-)								
T28	(-)	(-)					(x)				
T29	(-)						(-)				
T30		(-)					(-)				
T31						(x)		(x)			
T32	(-)						(-)				
T33									(x)	(-)	(-)
T34											
T35			(-)								
T36										(x)	
T37					(-)						
T38				(x)				(-)			
T39				(-)							(-)
T40											
T41			(-)		(-)					(x)	
T42	(x)	(x)					(-)				
T43	(-)	(-)					(x)				
T44				(-)					(x)		
T45											(-)
T46							(-)				
T47						(-)					
T48											
T49				(x)							(-)
T50								(x)	(-)		

Table A.36: Data and target assignment solution for problem P-n101 continued-4